TUGAS AKHIR – KI141502

RANCANG BANGUN APLIKASI WEB SERTA PENGAMANAN PADA SISTEM PEMUNGUTAN SUARA SECARA ELEKTRONIK (*E-VOTING*)

AHMAD NATSIR FAUZI
NRP 5111 100 160

Dosen Pembimbing
Prof. Ir. Supeno Djanali, M.Sc., Ph.D.
Henning Titi Ciptaningtyas, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2016

FINAL PROJECT – KI141502

WEB DESIGN APPLICATIONS AND SECURITY OF ELECTRONIC VOTING SYSTEM (E-VOTING)

AHMAD NATSIR FAUZI
NRP 5111 100 160

Supervisor
Prof. Ir. Supeno Djanali, M.Sc., Ph.D.
Henning Titi Ciptaningtyas, S.Kom., M.Kom.

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA 2015

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul "*Rancang Bangun Web Aplikasi serta Pengamanan Pada Sistem Pemungutan Suara secara Elektronik (*E-Voting*)*" dengan tepat waktu.

Harapan dari penulis, semoga apa yang tertulis di dalam buku Tugas Akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi yang nyata bagi kampus Teknik Informatika, ITS Surabaya, dan bangsa Indonesia.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Allah SWT, karena atas limpahan rahmat-Nya, penulis diberikan kemudahan dan kelancaran dalam mengerjakan Tugas Akhir ini.
2. Rasullullah Shalallahualaihi Wa Sallam atas doa kebaikan yang selalu beliau panjatkan untuk ummatnya.
3. Mama Lala dan Papa Muslich tercinta, Mas Aang, Mas Iput, Adek Risa, Adek Tia serta keluarga besar yang telah memberikan dukungan moral, spiritual, semangat, perhatian, selalu setia dan sabar dalam menghadapi curhatan dari penulis, serta selalu memberikan doa yang tiada habisnya yang dipanjatkan untuk penulis.
4. Prof. Ir. Supeno Djanali, M.Sc., Ph.D. dan Henning Titi Ciptaningtyas, S.Kom., M.Kom. selaku dosen pembimbing penulis yang telah memberikan banyak arahan, bantuan, koreksi, dan masukan yang membangun sehingga penulis dapat menyelesaikan Tugas Akhir ini.
5. Ibu Dr. Nanik Suciati selaku ketua jurusan Teknik Informatika ITS, dan segenap dosen Teknik Informatika yang telah memberikan ilmu kepada penulis.

`

6. Bapak dan Ibu staf Tata Usaha yang telah memberikan bantuan dan kemudahan kepada penulis selama masa perkuliahan di Teknik Informatika ITS.
7. Antek-antek sebagai grup penulis yang selalu memberikan semangat dan doa.
8. AYC sebagai teman yang selalu mendukung untuk menyelesaikan tugas akhir ini.
9. Pihak-pihak lain yang tidak bisa penulis sebutkan satu-persatu di sini yang juga telah membantu penulis menyelesaikan tugas akhir ini.

Seperti bunyi peribahasa, tiada gading yang tak retak, penulis menyadari sepenuhnya bahwa tugas akhir ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan saran dan kritik dari pembaca untuk perbaikan selanjutnya.

Surabaya, Januari 2016

Ahmad Natsir Fauzi

Penulis

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI WEB DAN KEAMANAN PADA SISTEM PEMUNGUTAN SUARA SECARA ELEKTRONIK (*E-VOTING*)

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Arsitektur Jaringan Komputer
Jurusan Teknik Informatika
Fakultas Teknologi Informasi

Oleh:

AHMAD NATSIR FAUZI
NRP 5111 100 160

Disetujui oleh Pembimbing Tugas Akhir:

Prof. Ir. Supeno Djanali, M.Sc., Ph.D. .....
NIP 19480619 197301 1 001 (Pembimbing 1)

Henning Titi C.,S.Kom., M.Kom. .....
NIP 19840708 201012 2 004 (Pembimbing 2)

KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
FAKULTAS TEKNOLOGI INFORMASI
JURUSAN TEKNIK INFORMATIKA

SURABAYA
JANUARI 2016

RANCANG BANGUN WEB APLIKASI DAN PENGAMANAN PADA SISTEM PEMUNGUTAN SUARA SECARA ELEKTRONIK (*E-VOTING*)

Nama	: Ahmad Natsir Fauzi
NRP	: 5111 100 160
Jurusan	: Teknik Informatika FTIf ITS Surabaya
Dosen Pembimbing	: Prof. Ir. Supeno Djanali, M.Sc., Ph.D. Henning Titi C., S.Kom., M.Kom.

Abstrak

Indonesia merupakan negara demokrasi dan menjunjung tinggi kedaulatan rakyat. Dimana untuk menentukan seorang pemimping dilakukan pemilihan secara serentak oleh masyarakat. Namun sistem pemungutan suara di Indonesia yang lebih dikenal dengan pemilihan umum (pemilu) masih dilaksanakan secara manual. dalam sistem tersebut, risiko kesalahan penghitungan suara tinggi karena banyaknya surat suara yang dihitung. Selain itu, rawan terjadi kecurangan terhadap jumlah suara demi kemenangan kelompok atau partai tertentu. Akibatnya, pelaksanaan pemilu menjadi tidak sesuai dengan asas yang berlaku. Untuk mengatasinya, dilaksanakan sistem pemilu modern yang disebut e-voting. Untuk beralih ke sistem e-voting ini sistem memerlukan aplikasi web yang dapat mengelola berbagai data yang diperlukan oleh sistem. Selain itu sistem harus dapat mengintegrasikan setiap data yang diperoleh dari setiap daerah di seluruh Indonesia serta dapat memberikan informasi dari sistem pemilihan yang dibuat secara modern .

Pada tugas akhir ini, penulis membuat sistem informasi berbasis web yang dapat mengelola berbagai data dari sistem pemilihan yang dibuat secara modern (e-voting) serta memberikan informasi hasil pemilihan secara modern.Untuk mengurangi resiko terjadi kecurangan penulis juga menerapkan beberapa sistem keamanan pada web aplikasi yang akan dibuat berupa CSRF Token pada setiap request yang dakan diakses oleh

`

web server baik get method atau post method. selain itu pengamanan juga dilakukan untuk melihat aktivitas admin dengan cara menyimpan session login admin.

Hasil yang dibuat oleh penulis terbukti sistem mampu mengolah beberapa data dan memberikan informasi berupa total data suara yang di tampilkan baik dalam bentuk tabel maupun grafik. Dari sisi keamanan, sistem yang telah dibuat terbukti aman dari beberapa serangan. Diharapkan sistem yang dibuat dapat diterapkan menuju sistem e-voting sebenarnya di masa mendatang.

Kata kunci: pemilu, e-voting, keamanan web, csrf token, session

Web Design Applications And Security Of Electronic Voting System (E-Voting)

Name : Ahmad Natsir Fauzi
NRP : 5111 100 160
Department : Informatics Engineering FTIf ITS Surabaya
Supervisor : Prof. Ir. Supeno Djanali, M.Sc., Ph.D.
Henning Titi C., S.Kom., M.Kom.

Abstract

Indonesia is a country based on democracy and highly uphold people sovereignty. And as a democracy country, to choose the lead of the country Indonesia hold the general election at the same time for the entire region of Indonesia. But, the voting system that used in the general election still depend on manual way. In that system, the risk in the fault of counting the voice of people is still high due to enermous ballot paper for voting to count. Beside that, the manual way of this voting will lead to several fraud of the ballot paper by some people for the advantage of certain group or political party. And the consequence is, the general election become not appropiate with election principle that applied in Indonesia. To cope with it, the best way is hold the voting system in modern way or known as e-voting. To move in this e-voting system, the system requires web application that can manage a big deal of massive data. Moreover, this e-voting system must be able to intergrate every data that received from every region in Indonesia and also be able to give informatiom from every received data.

In this last project, the author make a aystem information based on web that can manage various data from the election system made in modern way(e-voting). To decrease the risk of the fraud on electiom, the author also applying some security system on the application web that will be made as csrf tokken on every request which will be accessed by web server whether it get or

`

post method. Besides, the securement will be held for monitoring the activity of admin by saving log session in of administrator.

The results created by the author proved the system is able to process some of the data and provide information such as total voice data are displayed either in the form of tables and graphs. In terms of safety, proven system that has been made safe from attacks. The system is expected to be made can be applied to the actual e-voting system in the future.

Keyword: e-voting, web information system, csrf token session

DAFTAR ISI

`

DAFTAR GAMBAR

`

`

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Indonesia adalah negara yang menganut demokrasi dan menjunjung tinggi kedaulatan rakyatnya. Sarana pelaksanaannya adalah pemilihan umum; disingkat Pemilu. Tujuan dari pemilu itu sendiri adalah untuk memilih wakil rakyat yang akan duduk di parlemen. Pemilu yang diadakan setiap lima tahun sekali ini sangat menentukan bagi nasib bangsa Indonesia. Untuk itu berbagai cara agar pemilu ini dilakukan dengan efektif, terintegrasi serta jujur telah dicoba demi mendapatkan hasil yang lebih baik selama selama proses pemilihan umum berlangsung. Namun pelaksanaan Pemilu di Indonesia saat ini masih bersifat manual. Proses pemungutan dan penghitungan suara yang dilakukan masih bersifat manual dengan menggunakan kertas surat suara. Pelaksanaan semacam ini memiliki kelemahan-kelemahan, antara lain (1) tingginya risiko kesalahan penghitungan suara mengingat maki banyaknya jumlah surat suara yang harus dihitung dan (2) rawan terjadi kecurangan terhadap jumlah suara untuk kepentingan partai atau golongan tertentu. Akibatnya, kualitas pelaksanaan Pemilu dan kepercayaan rakyat terhadap hasil Pemilu menurun.

Salah satu solusi untuk mengatasi kelemahan-kelemahan tersebut adalah dengan melaksanakan sistem Pemilu yang lebih modern, yang dikenal dengan sebutan *e-voting* [1]. Dalam perancangannya, sistem e-voting harus mempunyai sistem informasi yang dapat mengelolah semua data yang berkaitan dengan pemilu secara aman. Selain itu sistem juga dapat mengintegrasikan data dari seluruh wilayah Indonesia.

Untuk beralih menggunakan sistem *e-voting*, sangat diperlukan kesiapan yang matang dari berbagai faktor pendukungnya, mulai dari rancangan sistem, infrastruktur, dan sumber daya manusia yang mumpuni.

Dalam tugas akhir ini, penulis akan mengimplementasikan sistem informasi berbasis web yang mendukung sistem pemilihan umum secara modern (e-voting). Dimana sistem ini nantinya dapat mengolah berbagai data yang dibutuhkan oleh sistem dan menampilkan hasil pemilihan berdasarkan wilayah di Indonesia dalam bentuk data maupun grafik. Selain itu sistem informasi ini diharapkan memenuhi aspek keamanan yaitu *integrity* yang menekankan bahwa informasi tidak boleh diubah tanpa seijin pemilik dan *Authentication* yang menyatakan bahwa informasi itu benar asli dari admin.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut :

1. Bagaimana membuat web aplikasi e-voting berdasarkan klasifikasi level admin ?
2. Bagaimana membuat laporan hasil pemilihan berdasarkan partai dan calon berdasarkan wilayah ?
3. Bagaimana mengamankan web aplikasi e-voting dari serangan luar maupun dalam?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, yaitu:

1. Keamanan berfokus pada aspek *integrity* dan *authentication*.
2. Data hasil pemilihan suara sudah dalam bentuk *cipher text*.
3. Sistem hanya mengolah data pemilih, panitia dan hasil suara partai maupun calon.

1.4 Tujuan

Tugas Akhir ini memiliki beberapa tujuan yang rinciannya adalah sebagai berikut.

1. Mengolah data keseluruhan dari sistem pemilu secara online

2. Memberikan informasi tentang keseluruhan hasil pemilu berdasarkan wilayah
3. Mempercepat proses pelaksanaan pemilihan umum secara online (e-voting)

1.5 Manfaat

Manfaat dari hasil pembuatan tugas akhir ini adalah:

1. Mendapatkan pemahaman implementasi asas Pemilu ke dalam sistem *e-voting*
2. Mendapatkan informasi tambahan tentang implementasi Pemilu dalam suatu aplikasi teknologi informasi.
3. Mempercepat proses perhitungan data hasil pemilu.

1.6 Metodologi

Tahap yang dilakukan untuk menyelesaikan Tugas Akhir ini adalah sebagai berikut.

1. **Penyusunan proposal tugas akhir**

   Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan ini terdiri dari hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu, dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Subbab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula subbab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

2. **Studi literatur**

   Pada tahap ini akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan implementasi, yaitu mengenai pelaksanaan Pemilu di Indonesia, sistem *e-voting*, *codeigniter*, *csrf token*, *highchart*, *datatabels.*

3. **Desain dan implementasi perangkat lunak**
   Pada tahap ini akan dilakukan implementasi algoritma yang telah direncanakan. Implementasi dilakukan dalam sistem berbasis *web* dengan bahasa pemrograman *php, javascript* dan *html*. Rencana kakas kerangka kerja dengan *codeigniter*.
4. **Uji coba dan evaluasi**
   Tahap ini merupakan tahap uji coba aplikasi dan evaluasi terhadap metode implementasi aplikasi sistem. Tahap uji coba dan evaluasi yang dilakukan berfokus pada sejauh mana penggunaan implementasi pengolahan data dan pengolongan level admin serta keamanan pada aspek *integrity* dan *authentication* .
5. **Penyusunan buku tugas akhir**
   Pada tahap ini dilakukan penyusunan Buku Tugas Akhir yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi yang telah dibuat.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini disusun dengan sistematika penulisan sebagai berikut:

1. Bab I. Pendahuluan
   Bab ini berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan Tugas Akhir.
2. Bab II. Tinjauan Pustaka
   Bab ini berisi penjelasan mengenai dasar teori yang mendukung pengerjaan Tugas Akhir.
3. Bab III. Analisis dan Perancangan
   Bab ini berisi penjelasan mengenai analisis kebutuhan, perancangan sistem dan perangkat yang digunakan dalam pengerjaan Tugas Akhir serta urutan pelaksanaan proses.
4. Bab IV. Implementasi

Bab ini berisi pembangunan aplikasi web aplikasi *e-voting* sesuai dengan rumusan dan batasan yang sudah dijelaskan pada bagian pendahuluan.

5. Bab V. Uji Coba dan Evaluasi

   Bab ini berisi pembahasan mengenai hasil dari uji coba yang dilakukan terhadap proses login admin berdasarkan level dan menampilkan data suara berdasarkan calon dan partai serta wilayah.

6. Bab VI. Kesimpulan dan Saran

   Bab ini berisi kesimpulan hasil penelitian. Selain itu, bagian ini berisi saran untuk pengerjaan lebih lanjut atau permasalahan yang dialami dalam proses pengerjaan Tugas Akhir.

BAB II
TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori yang berkaitan dengan implementasi perangkat lunak. Penjelasan tersebut bertujuan untuk memberikan gambaran sistem yang akan dibangun dan sebagai penunjang dalam pengembangan perangkat lunak.

2.1 E-Voting

Electronik voting (e-voting) adalah Pemilu yang memanfaatkan sarana teknologi informasi atau perangkat elektronik, di mana sebagian atau seluruh proses pelaksanaannya yang terdiri dari pendaftaran pemilih, pemungutan suara, hingga penghitungan suara, dilakukan secara digital. Banyak negara di dunia yang berhasil menerapkan e-voting meski memerlukan waktu yang cukup lama dalam mempersiapkannya, di antaranya adalah India dan Brazil. Di Indonesia sudah ada daerah yang berhasil menerapkan e-voting meski baru bisa diterapkan dalam lingkup Pilkades, yaitu Kabupaten Jembrana di Bali dan Kabupaten Boyolali di Jawa Tengah [2].

Kelebihan yang dimiliki e-voting antara lain adalah menghemat biaya pelaksanaan Pemilu, mempercepat proses pemungutan dan penghitungan suara, serta risiko kesalahan teknis yang kecil.

Penerapan e-voting dalam suatu negara memerlukan waktu yang cukup lama karena perlu adanya kesiapan yang matang dan menyeluruh terhadap faktor-faktor pendukungnya, mulai dari teknologi sistem yang digunakan hingga sumber daya manusia penggunanya. Untuk itulah, tujuan penggunaan e-voting tidak hanya sekadar mempercepat proses pemungutan dan penghitungan suara saja, akan tetapi yang lebih penting adalah untuk menjaga keaslian suara pemilih, kerahasiaan pemilih, dan juga menjaga akurasi penghitungan suara [3].

2.2 Fungsi Hash SHA-256

Fungsi hash adalah sebuah fungsi yang masukannya berupa string atau pesan dengan panjang sembarang dan keluarannya berupa nilai hash (hash value) atau pesan sidik dengan panjang tetap yang disebut digest/message digest. Fungsi hash dapat dituliskan dalam bentuk persamaan:

$$h = H(M)$$

di mana H merupakan fungsi hash yang menerima masukan berupa pesan M untuk menghasilkan keluaran berupa nilai hash atau digest h.

Fungsi hash dapat digunakan untuk mewujudkan layanan keamanan jaringan berupa keutuhan data (data integrity). Fungsi tersebut berguna sebagai parameter yang mengecek ada tidaknya perubahan terhadap data sebelum dan sesudah data itu dikirim atau disebar ke tempat lain dalam satu komputer ataupun melalui jaringan.

Salah satu cara untuk menghasilkan digest adalah dengan memanfaatkan penggunaan fungsi kompresi secara berulang terhadap sebuah pesan yang telah dibagi sebelumnya menjadi sejumlah blok pesan. Salah satu fungsi hash yang menggunakan cara tersebut adalah SHA (Secure Hash Algorithm).SHA adalah fungsi hash satu arah yang dibuat oleh National Institute of Standards and Technology (NIST). SHA didasarkan pada MD4 yang dibuat oleh Ronald L.Rivest. SHA sampai saat ini memiliki 4 versi, mulai dari SHA-0, SHA-1, SHA-2, dan SHA-3. SHA-2 memiliki beberapa varian, salah satunya adalah SHA-256.

SHA-256 adalah fungsi hash SHA yang menghasilkan nilai hash dengan panjang 256 bit dengan menggunakan iterasi terhadap sejumlah blok pesan yang masing-masing panjangnya 512 bit.

Secara singkat, langkah-langkah untuk menghasilkan digest adalah sebagai berikut. Misalkan ada pesan M. M tersebut dicari representasi biner per karakter pesannya. Berikutnya adalah menambah padding terhadap M berupa satu bit karakter "1", diikuti sejumlah bit karakter "0" dan representasi biner dari

panjang byte pesan, agar panjang M tepat berkelipatan 512 bit. Representasi pesan M ber-padding tersebut kemudian diubah menjadi bilangan heksadesimal. Lalu, memanggil fungsi kompresi dengan parameter masukan berupa delapan word nilai digest awal yang diambil dari bagian pecahan pada akar kuadrat delapan bilangan prima pertama dengan panjang word masing-masing 32 bit dan pesan M ber-padding. Dalam fungsi kompresi, dilakukan ekspansi word dari pesan M ber-padding tersebut dengan jumlah word sebanyak 64 yang masing-masing panjangnya 32 bit dan komputasi ronde sebanyak 64 ronde, yang mana hasilnya didapatkan dengan melakukan penjumlahan digest dikalikan dengan mod 232. Setelah tidak ada blok pesan lagi yang diproses, maka hasil dari fungsi kompresi dijadikan sebagai digest yang dicari dengan panjang 256 bit [4].

Pada Tugas Akhir ini metode hash SHA256 digunakan untuk memetakan string berupa password yang akan dieksekusi oleh sistem yang bertujuan mencocokkan string password admin dengan password yang di telah disimpan di database.

2.3 PHP

Kode PHP dapat hanya digabung dengan kode HTML, atau dapat digunakan dalam kombinasi dengan berbaga template dan kerangka kerja web. Kode PHP biasanya diproses oleh interpreter PHP, yang biasanya diimplementasikan sebagai modul asli web server atau *Common Gateway Interface (CGI) executable*. Setelah kode PHP ditafsirkan dan dilaksanakan, server web mengirimkan output yang dihasilkan untuk klien, biasanya dalam bentuk bagian dari halaman web yang dihasilkan; misalnya, kode PHP dapat menghasilkan halaman Web kode HTML, gambar, atau data lainnya. PHP juga telah berkembang untuk menyertakan antarmuka baris perintah (CLI) kemampuan dan dapat digunakan dalam aplikasi grafis[5].

Pada Tugas Akhir ini kode PHP digunakan pada keseluruhan sistem yang dibangun untuk mengimplementasikan beberapa fungsi yang bertujuan untuk menampilkan informasi.

2.4 CSRF(*Cross-Site Request Forgery*) Token

Cross-Site Request Forgery adalah jenis serangan berbahaya yang terjadi pada sebuah situs Web, email, blog, pesan instan, atau program menyebabkan pengguna web browser untuk melakukan tindakan yang tidak diinginkan pada situs yang terpercaya ketika pengguna sudah dikonfirmasi. Dampak dari serangan CSRF terbatas pada kemampuan menampilkan kerentanan pada aplikasi web server. Misalnya, serangan ini melakukan transfer dana, mengubah password, atau membeli item dalam konteks pengguna. Akibatnya, serangan CSRF digunakan oleh penyerang untuk membuat sistem target melakukan fungsi (dana transfer, pengiriman formulir dll) melalui browser target tanpa pengetahuan tentang target pengguna, setidaknya sampai fungsi yang tidak sah telah dilakukan.

Salah satu untuk menangkal serangan csrf ini adalah dengan menambahkan token pada permintaan baik method *GET* atau *POST* yang dilakukan oleh server. CSRF token adalah metode penambahan token yang dilakukan oleh client yang akan mengeksekusi permintaan *GET* atau *POST*, token ini biasanya berisi *session* yang dienkripsi dalam bentuk hash[6].

CSRF token digunakan sebagai salah satu implementasi dari pengamanan untuk mengamankan sistem dari bentuk serangan CSRF pada sistem informasi web e-voting yang akan dibuat.

2.5 CodeIgniter

CodeIgniter merupakan aplikasi sumber terbuka yang berupa framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP. CodeIgniter memudahkan developer untuk membuat aplikasi web dengan cepat mudah dibandingkan dengan

membuatnya dari awal. CodeIgniter dirilis pertama kali pada 28 Februari 2006.

Model View Controller merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman Small Talk, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, user interface, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC pattern dalam suatu aplikasi yaitu :

View, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller. View berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.

Model, biasanya berhubungan langsung dengan database untuk memanipulasi data *(insert, update, delete, search)*, menangani validasi dari bagian controller, namun tidak dapat berhubungan langsung dengan bagian view.

Controller, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view, controller berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.

Dengan menggunakan prinsip MVC suatu aplikasi dapat dikembangkan sesuai dengan kemampuan developernya, yaitu programmer yang menangani bagian model dan controller, sedangkan designer yang menangani bagian view, sehingga penggunaan arsitektur MVC dapat meningkatkan maintanability dan organisasi kode. Walaupun demikian dibutuhkan komunikasi yang baik antara programmer dan designer dalam menangani variabel-variabel yang akan ditampilkan[7].

Pada sistem yang akan dibuat, sistem menggunakan framework codeigniter, sistem juga mengimplementasikan beberapa fitur dari codeigniter.

2.6 Highcharts

Highcharts adalah *library charting* (pustaka grafik) yang ditulis dalam JavaScript murni, highchat memberi cara yang muduh untuk menambahkan interaktif grafik kedalam situs web atau aplikasi web. Highchart mendukung grafik yang berupa garis, spline, areaspline, kolom, *bar*, *pie*, pencar, alat pengukur sudut dan lain-lain[8].

Pada sistem yang akan dibuat, highchart digunakan untuk mempresentasikan hasil penghitungan suara oleh sistem pemilu secara online. Contoh dari grafik yang dibuat oleh highchart yang ditunjukkan pada Gambar 2.1.

Gambar 2.1 Contoh Highcharts Grafik

BAB III
DESAIN DAN PERANCANGAN

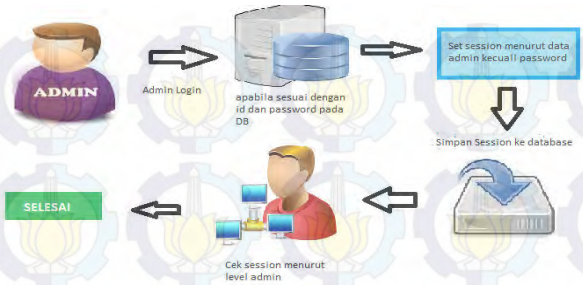
Bab ini berisi pembahasan mengenai perancangan sistem perangkat lunak untuk mencapai tujuan dari Tugas Akhir. Perancangan yang akan dijelaskan pada bab ini meliputi perancangan data, perancangan basis data, dan perancangan antarmuka. Selain itu akan dijelaskan juga desain metode secara umum pada sistem.

3.1 Perancangan Sistem secara Umum

Pada Tugas Akhir ini akan dibangun aplikasi berbasis web untuk mendukung jalanya sistem pemungutan suara secara online (e-voting). Dimana nantinya admin akan mengolah data dari seluruh data yang dibutuhkan oleh sistem pemilihan umum secara online (e-voting) namun untuk alasan keamanan data hasil pemungutan suara tidak boleh diolah oleh admin termasuk menghapus ataupun merubah, admin hanya berhak untuk menampilkan data suara baik berupa tabel maupun berupa grafik. Admin akan digolongkan berdasarkan level admin, alur yang dijalankan sistem untuk pengecekan level admin adalah pertama kali admin melakukan login berdasarkan Id dan password, dalam hal ini password di hash menggunakan metode hash SHA-256, jika cocok sistem akan melakukan penerapan sesi dengan data admin kecuali password kemudian session akan melakukan pengecekan sesi level admin yang di simpan dalam tabel session. Kemudian sistem akan mengolah data berdasarkan kebutuhan dari sistem pemilu yang akan berlangsung. Dan setiap aktivitas yaitu menambah, merubah dan menghapus data yang dilakukan oleh setiap level admin akan dicatat semua kedalam log database yang itu dapat dilihat pada level diatasnya hal ini bertujuan untuk memonitoring admin agar tidak terjadi kecurangan pada sisi dalam sistem yaitu admin itu sendiri.

Gambar 3.1 merupakan alur dari pengecekan level admin oleh sistem.

.



Gambar 3.1 Alur Login Admin Pada Sistem

ada 4 level dalam penggolongan admin pada sistem yang akan dibuat.

1. Admin level Kota/Kabupaten

   Admin level Kota/Kabupaten mempunyai wewenang untuk mengelolah data diseluruh Kota/Kabupaten. Adapun beberapa kewenangan dari admin Kabupaten/Kota.

   a) *CRUD(create, update, delete)* data panitia
   b) Mengubah dan Memasukkan data pemilih
   c) Memvalidasi data pemilih. Dalam hal ini apakah pemilih masih dinyatakan aktif atau tidak aktif.
   d) Menambahkan admin level Kecamatan, Desa/Kelurahan dan TPS
   e) Merekap data hasil pemungutan suara baik suara partai maupun calon partai dalam bentuk grafik dan tabel.
   f) Backup Database

2. Admin level Kecamatan

   Admin level Kecamatan mempunyai wewenang untuk mengelolah data diseluruh Kecamatan. Adapun beberapa kewenangan dari admin Kabupaten/Kota.

a) *CRUD(create, update, delete)* data panitia
b) Mengubah dan Memasukkan data pemilih
c) Memvalidasi data pemilih. Dalam hal ini apakah pemilih masih dinyatakan aktif atau tidak aktif.
d) Menambahkan admin level Desa/Kelurahan dan TPS
e) Merekap data hasil pemungutan suara baik suara partai maupun calon partai dalam bentuk grafik dan tabel.
f) Backup Database

3. Admin level Desa/Kelurahan

Admin level Desa/Kelurahan mempunyai wewenang untuk mengelolah data diseluruh Desa/Kelurahan. Adapun beberapa kewenangan dari admin Kabupaten/Kota.

a) *CRUD(create, update, delete)* data panitia
b) Mengubah dan Memasukkan data pemilih
c) Memvalidasi data pemilih. Dalam hal ini apakah pemilih masih dinyatakan aktif atau tidak aktif.
d) Menambahkan admin level TPS
e) Merekap data hasil pemungutan suara baik suara partai maupun calon partai dalam bentuk grafik dan tabel.
f) Backup Database

4. Admin level TPS

Admin level TPS mempunyai wewenang untuk mengelolah data diseluruh TPS. Adapun beberapa kewenangan dari admin Kabupaten/Kota.

a) Mengubah dan Memasukkan data pemilih
b) Memvalidasi data pemilih. Dalam hal ini apakah pemilih masih dinyatakan aktif atau tidak aktif.

c) Merekap data hasil pemungutan suara baik suara partai maupun calon partai dalam bentuk grafik dan tabel.

d) Backup Database

5. Admin KPU

Admin KPU merupakan admin yang mempunyai level paling tinggi, tugas admin KPU adalah membuka data suara partai serta untuk melihat aktivitas semua level admin di berbagai level wilayah. Admin KPU juga mempunyai wewenang untuk merubah status *lock* admin yang terkunci.

Gambar 3.2 Menjelaskan beberapa dari hak akses admin menuut klasifikasi level admin.

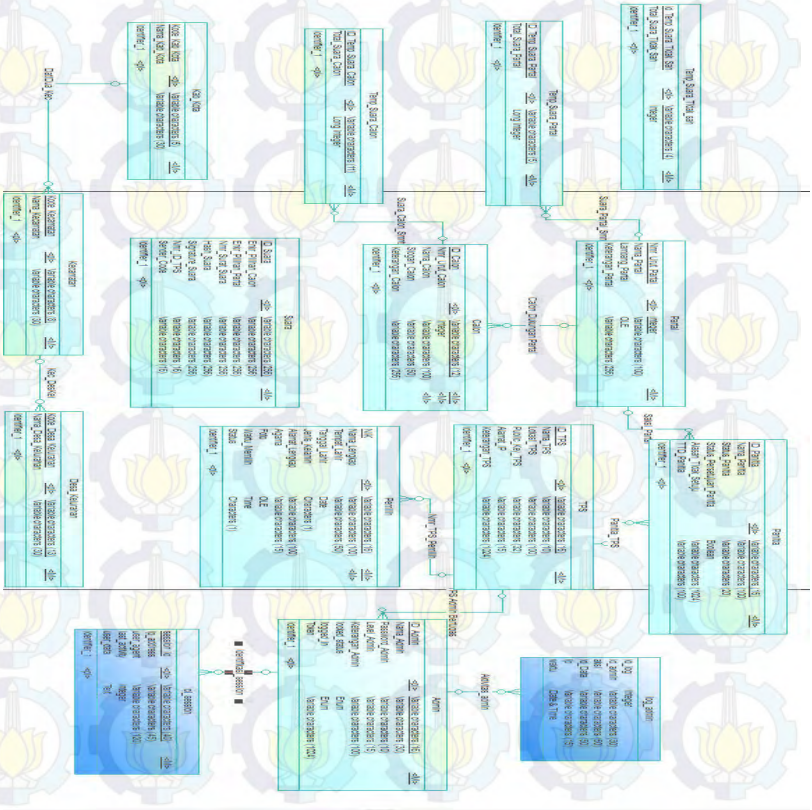
Gambar 3.2 Hak Akses Admin

3.2 Perancangan Basis Data

Perancangan basis data pada sistem yang akan dibuat berisi perancangan pdm dan beberapa tabel yang ditambahkan oleh penulis pada tabel sebelumnya.

3.2.1 Perancangan PDM

Perancangan basis data ini merujuk pada basis data yang sudah dibuat sebelumnya yang telah dilampirakan pada halaman lampiran. Namun ada beberapa pembaruhan pada sebagian tabel yang ditunjukkan pada Gambar 3.3 yang diberi warna berbeda.



Gambar 3.3 ***Physical Data Model*** **Sistem** ***Web aplikasi E-voting***

3.2.2 Perancangan Tabel-Tabel dalam Basis Data beserta Penjelasannya

Ada beberapa penambahan tabel maupun atribut yang ditambahkan pada tabel – tabel yang dirancang sebelumnya. Hal ini dikarenakan masih ada tabel yang belum lengkap. Masing-masing tabel dan atribut yang terdapat dalam basis data tersebut akan dijelaskan di bawah ini. Untuk diperhatikan bahwa atribut bertanda *) merupakan kunci utama/*primary key* dan **) merupakan kunci tamu/*foreign key* yang merujuk pada atribut milik tabel lain.

Tabel ini berfungsi untuk menyimpan data pemilih.

- Tabel Pemilih

Tabel 3.1 Penjelasan Atribut-Atribut Tabel Pemilih

Nama Atribut/Field	Tipe Data	Keterangan
NIK*)	varchar(16)	Nomor ID pemilih
ID_TPS**)	varchar(16)	Nomor ID TPS
Nama_Lengkap	varchar(100)	Nama lengkap pemilih
Tempat_Lahir	varchar(50)	Tempat lahir pemilih
Tanggal_Lahir	date	Tanggal lahir pemilih
Jenis_Kelamin	char	Jenis kelamin pemilih
Alamat_Lengkap	varchar(100)	Alamat lengkap pemilih
Agama	varchar	Agama pemilih
Foto	blob	Foto diri pemilih
Waktu_Memilih	time	Sebagai *flag*, penanda bahwa pemilih sudah memilih atau belum
Status	Varchar(1)	Sebagai penanda bahwa pemilih masih aktif

- Tabel Admin
  Tabel ini berfungsi untuk menyimpan data admin TPS.

Tabel 3.2 Penjelasan Atribut-Atribut Tabel Admin

Nama Atribut/Field	Tipe Data	Keterangan
ID_Admin*)	varchar(16)	Nomor ID admin
ID_TPS**)	varchar(16)	Nomor ID TPS
Nama_Admin	varchar(30)	Nama admin
Password_Admin	varchar(10)	Sandi masuk admin
Level_Admin	varchar(15)	Level admin (TPS, desa/kelurahan, kecamatan, atau kabupaten/kota)
Keterangan_Admin	varchar(100)	Info tambahan mengenai admin
Locked_status	Enum	Sebagai flag tanda untuk mengecek apakah id sedang dikunci, dalam hal ini karena gagal login selama 3kali
Logged_in	Enum	Sebagai flag penanda apakah admin sedang login ata tidak.
Token **)	Integer	Sebagai parameter untuk mencocokkan attribut dengan session id.

- Tabel ci_session
  Tabel ini berfungsi untuk menyimpan session dari data admin dan waktu aktivitas admin. Tabel ini juga berfungsi sebagai parameter untu mengontrol percobaan login admin.

Tabel 3.3 Penjelasan Atribut-Atribut CI Session

Nama Atribut/Field	Tipe Data	Keterangan
Session_id*)	varchar(40)	Menyimpan sesi dari suatu state ketika admin login dalam bentuk random hash
Ip_addresss	varchar(16)	Menyimpan ip address admin
User_agent	varchar(120)	Menyimpan user agent data,
User_data	Text	Menyimpan data admin
Last_activity	Integer	Menyimpan waktu aktivitas admin ketika login dalam bentuk timestamp
Loggin_attempt	Integer	Menyimpan percobaan lgoin admin
Token	Integer	Sama dengan pada tabel admin

- Tabel log_admin
  Tabel ini berfungsi untuk mencatat aktivitas dari admin baik menambah, merubah dan menghapus data – data pada web evoting sehingga memudahkan admin untuk melakukan pengecekan aktivitas admin.

Tabel 3.4 Penjelasan Atribut-Atribut Log_admin

Nama Atribut/Field	Tipe Data	Keterangan
Id_log*	varchar(15)	Menyimpan id log admin ketika melakukan aksi
Id_admin**	varchar(30)	Menyimpan id admin
Aksi	varchar(50)	Menyimpan aksi yang dilakukan oleh admin
Ip	Varchar(15)	Menyimpan data ip admin
Id_data	Varchar(100)	Menyimpan id data yang dieksekusi oleh admin ketika melakukan aksi
Waktu	datetime	Menyimpan ketika admin melakukan sesuatu

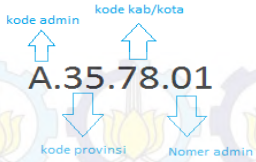
3.3 Perancangan Data

Data yang digunakan dan diproses dalam aplikasi sistem ini adalah Id Admin, Id TPS dan Id Calon yang digunakan untuk memudahkan hasil penyaringan.

3.3.1 Rancangan Data Kode Id Admin

Id admin merupkan salah satu atribut dari tabel admin, kode id admin merupakan atribut yang bersifa unik, Id admin dikodekan berdasarkan dari kode wilayah id Indonesia. Setiap id admin mempunyai perbedaan panjang karakter berdasarkan level admin tersebut. Hal ini untuk memudahkan penyaringan data ang dibutuhkan oleh sistem pemilihan umum e-voting. Berikut adalah penjelasan penkodean Id Admin.

- Untuk admin level Kabupaten/Kota pengkodean ditunjukkan seperti pada Gambar 3.4

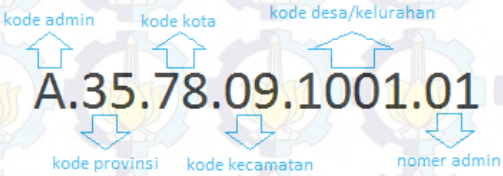


Gambar 3.4 Pengkodean Pada Admin Kab/kota

- Untuk admin level kecamatan pengkodean ditunjukkan seperti pada Gambar 3.5

Gambar 3.5 Pengkodean Pada Admin Kecamatan

- Untuk admin level Desa/Kelurahan pengkodean ditunjukkan seperti pada Gambar 3.6



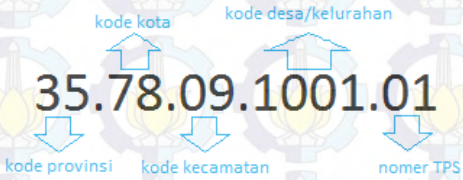
Gambar 3.6 Pengkodean Pada Admin Desa

- Untuk admin level TPS pengkodean ditunjukkan seperti pada Gambar 3.6

Gambar 3.7 Pengkodean Pada Admin TPS

3.3.2 Rancangan Kode Atribut Id TPS

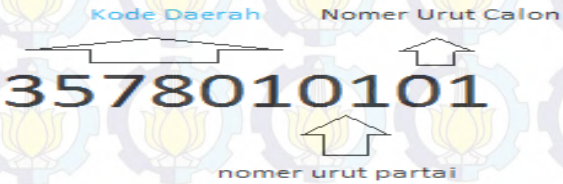
Pengkodean atribut Id TPS pada dasarnya mempunyai fungsi yang sama yaitu untuk memudahkan penyaringan TPS berdasarkan wilayah TPS berada. Adapun pengkodeannya telah di contohkan pada Gambar 3.8 .



Gambar 3.8 Pengkodean Id TPS

3.3.3 Rancangan Kode Atribut Id Calon

Berdasarkan pemilihan umum yang berlaku di Indonesia, terdapat 3 tingkatan dalam pemilihan wakil rakyat baik tingkat DPRD kabupaten/kota, DPRD Provinsi, dan DPR pusat yang berskala nasional untuk itu ID admin perlu dikodekan menurut tingkatan tersebut. Adapun contoh pengkodean Id calon yang akan dibuat adalah seperti pada Gambar 3.9.



Gambar 3.9 Pengkodean Id Calon

3.4 Perancangan Keamanan Pada Sistem

Sistem web *e-voting* dirancang untuk sebisa mungkin aman baik itu keamanan dari sisi luar yaitu penyerang maupun dalam yaitu admin itu sendiri. Untuk itu beberpa metode pengamanan akan diterapkan pada sistem yang akan dibuat. Beberapa rancang keamanan yaitu.

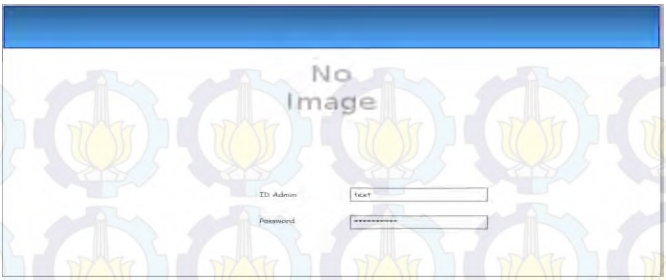
- Sistem Menggunakan HTTPS(*Hypertext Transfer Protocol Secure*).
- Password yang digunakan oleh admin menggunakan SHA-256.
- Mengaplikasikan percobaan login sebanyak 3 kali.
- Menggunakan *captcha* pada form login
- Menggunakan *XSSclean* pada *input text* login dan menambahkan *CSRFtoken* pada setiap permintaan oleh server
- Menambah tabel log admin yang digunakan untuk memantau aktivitas admin.

3.5 Perancangan Antarmuka Aplikasi

Antarmuka aplikasi dirancang untuk mempermudah interaksi antara aplikasi dengan pengguna. Pengguna di sini adalah seorang admin beberapa halaman dedesian sama pada setiap level admin baik itu level kota, kecamatan, desa namun ada beberapa halaman yang berbeda yang untuk halaman admin. Aplikasi ini memiliki beberapa halaman utama yaitu.

3.5.1 Halaman Login

Halaman ini merupakan halaman yang akan muncul ketika pertama kali aplikasi dijalankan. Jendela ini digunakan untuk *login* admin sehingga admin masuk kedalam sistem.. Gambar 3.10 menunjukkan rancangan jendela beranda.



Gambar 3.10 Rancangan Halaman Login

3.5.2 Halaman *Dashboard* Admin

Halaman ini merupakan *dashbord* dari admin atau halaman utama admin setelah admin berhasil login semua level admin mempunyai antar muka yang sama yang berisi informasi admin dan tombol validasi dan rekap partai. Dan di bagian sidebar terdapat fitur - fitut utama fungsionalitas pada admin atau hak yang dapat diakses oleh admin tiap level daerah menampilkan data panitia, pemilih, tambah admin rekap partai, statistik suara partai dan *backup* database partai. Untuk admin level TPS pada halaman sidebar berbeda dengan admin lainya karena admin level TPS hanya mempunyai akses untuk mengolah data pemilih dan suara yang akan ditampilkan.

Gambar 3.11 Rancangan Halaman Dashboard Admin

3.5.2.1 Halaman Validasi Pemilih

Halaman ini berisi tabel dari data pemilih menurut level wilayah admin. Terdapat *checkbox* dan beberapa tombol untuk memvalidasi admin.

Gambar 3.12 Rancangan Halaman Validasi

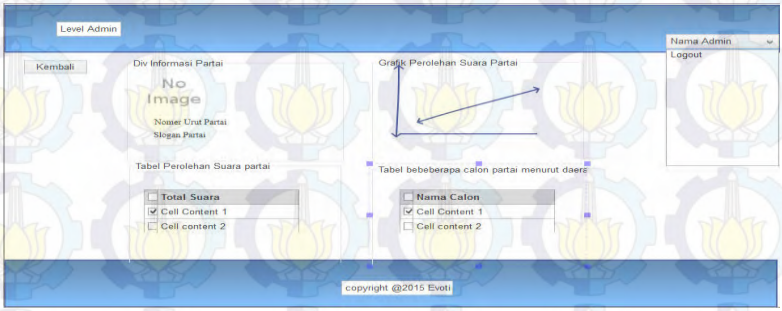
3.5.3 Halaman Rekap Partai

Halaman ini merupakan rekap data suara partai, pada halaman ini ada beberapa gambar partai yang terdaftar pada peserta pemilihan umum pada setiap gambar partai terdapat tautan untuk detail informasi dari partai serta beberapa data calon yang merupakan calon dari partai tersebut.

Gambar 3.13 Rancangan Halaman Rekap Partai

3.5.3.1 **Halaman Detail Partai**

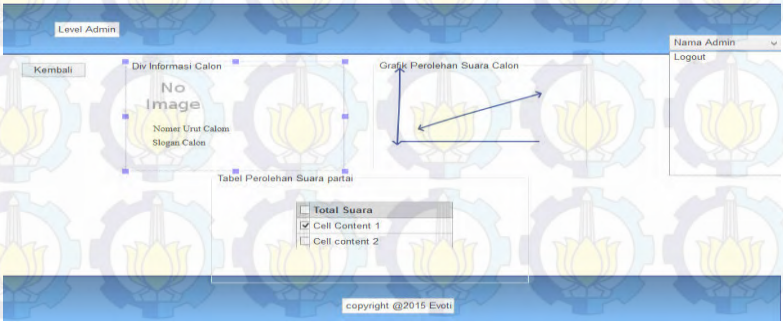
Pada Halaman ini terdapat detail data partai, yaitu suara masuk partai per TPS menurut level wilayah dan total seluruh suara partai serta grafik perolehan partai dilihat dari perbandingan suara partai dan seluruh suara yang masuk. Termasuk juga data calon yang berasal dari partai tersebut menurut daerah pemilihan. Pada halaman deatail juga terdapat tombol detail calon.



Gambar 3.14 Rancangan Halaman Detail Partai

3.5.3.1.1 **Halaman Detail Calon**

Pada halaman detail calon, menampilkan beberapa detail informasi calon partai, dan suara yang masuk berdasarkan level tiap daerah dalam bentuk tabel maupun grafik.



Gambar 3.15 Rancangan Halaman Detail Calon

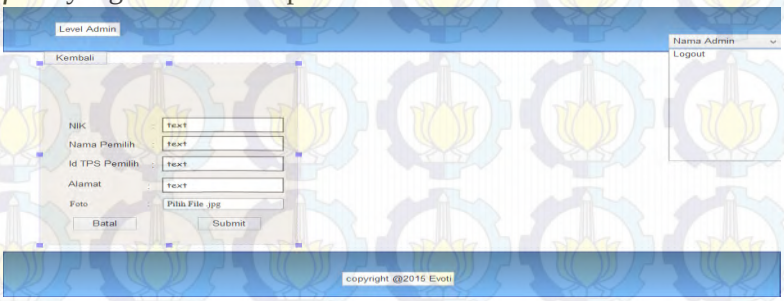
3.5.4 Halaman Data Pemilih

Di dalam halaman data pemilih terdapat tabel dari rekap semua data yang pemilih. Dalam halaman tersebut terdapat tombol yang berfungsi untuk kembali ke halaman *dashborad*, menambah data pemilih, merubah dan melihat detail dari data pemilih.

Gambar 3.16 Rancangan Halaman Data Pemilih

3.5.4.1 Halaman Update dan Add Data Pemilih

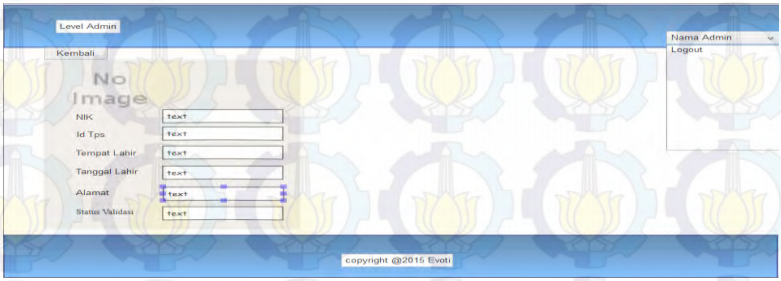
Halaman *update* dan *add* data pemilih berisi form yang digunakan untuk mengupdate data atau menambahkan data pemilih, method yang digunakan untuk permintaan server adalah *post*, pada method ini *CSRFtoken* akan diaplikasikan pada *hidden form* yang akan dikirim pada server.



Gambar 3.17 Rancangan Halaman Form Add Pemilih

3.5.4.2 Halaman Detail Data Pemilih

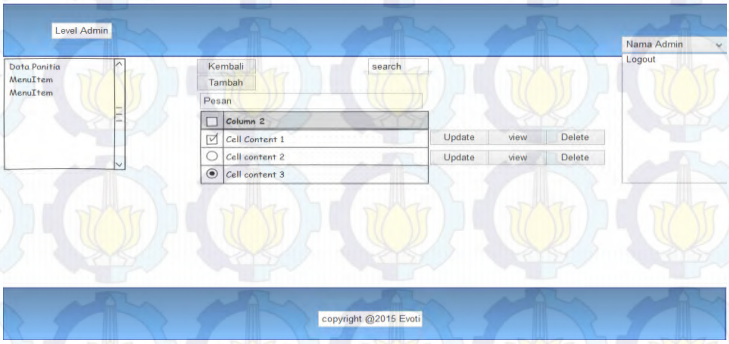
Halaman detail pemilih berisi dari detail pemilh yang akan dipilih pada halaman data admin, halaman detail pemilih menampilkan atribut – atribut yang tidak ditampilkan pada halaman data pemilih



Gambar 3.18 Rancangan Halaman Detail data Pemilih

3.5.5 Halaman Data Panitia

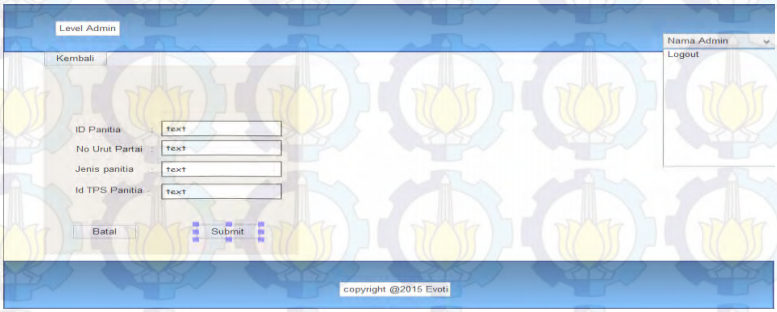
Pada halaman ini ditampilkan data panitia baik itu KPPS maupun Saksi. Halaman ini terdapat tombol untuk menambah, merubah, menghapus serta melihat secara detail data panitia tersebut.



Gambar 3.19 Rancangan Halaman Data Panitia

3.5.5.1 Halaman Update dan Add Panitia

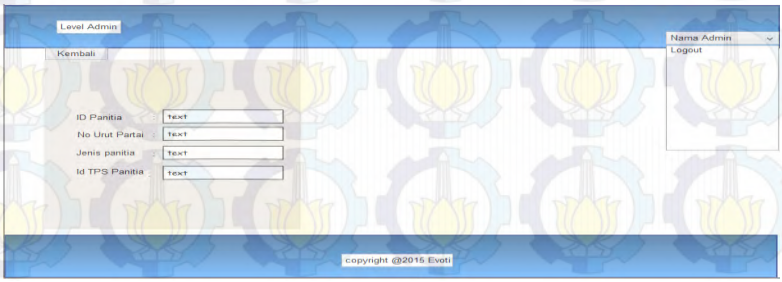
Pada halaman ini berisi form untuk menambahkan atau merubah data panitia. Dalam halaman ini terdapat fungsi untuk pengecekan uri apabila uri adalah *add* maka *textbox* mempunyai nilai kosong, dan apabila uri adalah update maka otomatis *textbox* berisi nilai dari masing – masing atribut data panitia. Untuk pengamanan terdapat form yang tersebunyi yang berisi token untuk menangkal serangan csrf token.



Gambar 3.20 Rancangan Halaman Form add/update Panitia

3.4.2.2 Halaman Detail data Panitia

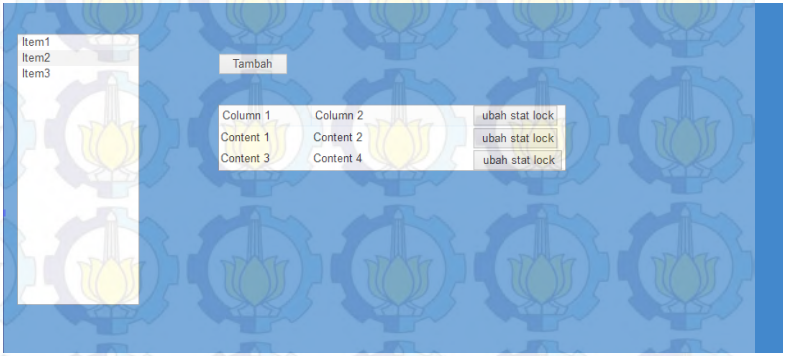
Pada halaman detail panitia berisi tentang data detail panitia yang tidak ditampilkan pada tabel. Serta untuk mengtahui tentang status panitia mengenai data suara yang masuk untuk apakah panitia sudah setuju atau tidak.



Gambar 3.21 Rancangan Halaman Form add Pemilih

3.5.6 Halaman Data Admin

Pada halaman data admin terdapat tabel admin yang berisi admin level dibawahnya, misalnya apabila login sebagai admin Kabupaten/Kota maka halaman admin hanya bisa menampilkan admin kecamatan dan seterusya, terdapat tombol untuk merubah status *lock* admin pada bagian kanan tabel. Gambar 3.22 merupakan rancangan data admin yang akan dibuat.



Gambar 3.22 Halaman Admin

BAB IV
IMPLEMENTASI

Bab ini berisi pembahasan mengenai implementasi antarmuka aplikasi berdasarkan perancangan yang telah dibuat. Tahap perancangan merupakan tahap dasar dari implementasi perangkat lunak.

4.1 Lingkungan Implementasi

Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam implementasi ini ditampilkan pada Tabel 4.1 berikut.

Tabel 4.1 Lingkungan Perancangan Perangkat Lunak

Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel® Core™ i5-3337U CPU @ 1.80 GHz Memori: 4.00 GB
Perangkat lunak	Sistem Operasi: Microsoft Windows 7 Home Premium 32-bit Perangkat Pengembang: Codeigniter version 2.1.0 Perangkat Pembantu: PHPMyAdmin, Sublime, Microsoft Word 2013, IIS 8.5 Pustaka: Php,mySqldata,JavaScript

4.2 Implementasi Penggolongan Level pada sistem

Ada 4 macam penggolongan level admin yang sudah dijelaskan pada bab 3. Penggolongan ini diaplikasikan ketika admin berhasil login dan data Admin disimpan dalam database session. Kemudian untuk setiap fungsi pada level admin akan

dilakukkan cek sesi berdasarkan penggolongan admin. Berikut adalah rancang kode yang dijelaskan pada Gambar 4.1.

```
 function set session($username)
   {
        $this->session->set_userdata($username);
        $this->session->set_userdata('logged_in',true);

            }
    $cek="";
    function cek_session()
    {
       $this->cek = $this->session->userdata("LEVEL ADMIN");
        return $this->cek;
    }
    if ($result === true) {
                if($this->cek_session() == "Kabupaten/Kota")
                    {
                        redirect("admin kota");

                    }
                    elseif ($this->cek_session() ==
"Kecamatan") {
                        redirect("admin kecamatan");
                    }
                    elseif ($this->cek session() =="TPS") {
                        redirect("admin_tps");
                    }
                    elseif ($this->cek_session()
=="Desa/Kelurahan") {
                        redirect("admin desa");
                    }
                }
            else {
```

Gambar 4.1 Gambar Rancang Kode Penggolongan Admin

4.3 Implementasi *CSRF token* dan *XSSClean*

Untuk manangani serangan *CSRF*, *framework* codeigniter menyediakan fungsi untuk membuat token yang berisi random string setiap pengguna melakukakan request pada server atau yang disebut *CSRF token*. *CSRF token* ini akan aktif apabila pada config.php pada codeigniter disetting menjadi *true* seperti pada Gambar 4.2.

```
$config['csrf_protection'] = TRUE;
$config['csrf_token_name'] = 'csrf_test_name';
$config['csrf_cookie_name'] =
'csrf_cookie_name';
$config['csrf_expire'] = 7200;
```

Gambar 4.2 Rancang Kode Set *CSRFtoken*

Secara *default* untuk membuat *form method* menggunakan *form helper* yang ada pada codeigniter *csrf token* akan *digenerate* secara otomatis, apabila menggunakan http form method biasa maka akan dibuat secara manual seperti pada Gambar 4.3

```
form_hidden($this->security-
>get_csrf_token_name(), $this->security-
>get_csrf_hash());
```

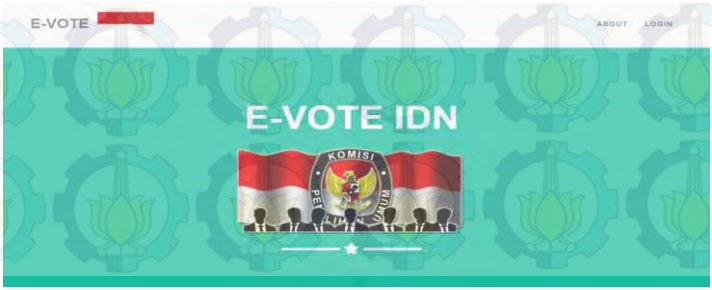
Gambar 4.3 *Generate CSRFtoken*

4.4 Implementasi Antarmuka Aplikasi

Implementasi antar muka aplikasi terdiri dari beberapa halaman utama yaitu halaman login dan halaman utama, halaman utama terdiri dari beberapa sub halaman yang akan dijelaskan pada sub bab berikut.

4.4.1 Implementasi Halaman Login Aplikasi

Gambar 4.4 menunjukkan implementasi halaman login aplikasi. Yang merupakan tampilan awal pada web ketika di akses.



Gambar 4.4 Implementasi halaman login aplikasi

4.4.1.1 Implementasi Tombol Login

Tombol Login berfungsi untuk mengizinkan admin masuk ke jendela utama aplikasi. Setelah tombol ini diklik, maka sistem akan mengecek admin, apakah Id dan Password admin yang digunakan cocok dengan data dalam database. Pada sistem login terdapat beberapa alur pengecekan. Petama ada validasi dari input yaitu id dan password apakah sesuai dengan aturan string yang diset dan beberapa pengecekan karakter untuk mengurangi kerentanan serangan *xss(cross x scripting)*. Yang kedua pengecekan Id dan password, disini menggunakan metode hash sha256 untuk mengacak string password maka untuk mengecek apakah sama dengan data admin menggunakan decode SHA 256. Kemudian jika id cocok dengan yang ada pada database sistem akan menyimpan token dari data admin sebagai session. Dan melakukan iterasi untuk setiap percobaan login. Apabila admin salah memasukkan password selama 3 kali maka secara otomatis akun akan di blokir. Dan jika berhasil maka admin akan dicek menurut level admin. Pada bab perancangan sudah dijelaskan ada 4 penggolongan admin menurut level wilayah. Maka admin akan masuk berdasarkan levelnya. Gambar 4.5 merupakan form dari login.

Gambar 4.5 Form Login

Gambar 4.6 Merupakan beberapa rancangan kode controller yang mengimplementasikan fungsi untuk mengecek id admin dan password.

```
            $name = $this -> input -> post('userid');
            $password = $this -> input -> post('password');
            //load model
            $this -> load -> model('login model');
            $result = $this -> login_model ->
get_user($name, $password);

            if ($result === false) {
                $data['error message'] = "Login gagal.
ID atau Password Salah!!!";
                $data['title'] = "Login Page";
                                $this -> load ->
view('login_view', $data);


            } else if ($result === 'locked') {

                $data['error_message'] = "Untuk
sementara akun anda diblokir</a>";
                $data['disabled'] = "disabled";
                $data['signin'] = anchor('admin/logout',
'Coba login dengan akun lain');
                $data['title'] = "Login Page";
                $this -> load -> view('login_view',
$data);
```

Gambar 4.6 Rancang Kode dari Controller Login

Gambar 4.7 merupakan implementasi dari model login yang mempunyai fungsi untuk mencocokkan Id Admin dan Password admin dan mengolah session admin yang akan login.

```
            $idadmin = $query['ID_ADMIN'];
            $user_password = $query['PASSWORD_ADMIN'];
            $id_tps = $query['ID_TPS'];
            $level = $query['LEVEL_ADMIN'];
            $nama = $query['NAMA_ADMIN'];
$password = hash('sha256',$password);

            if ($password != $user_password) {
```

```
                $session id = $this-> session ->
userdata('session id');
                return ($this ->
update ci session($session id, $token));
            }
            else if (($password === $user_password)) {

    function update_ci_session($session_id, $token) {
            $attempt = $this->session-
>userdata['login attempt'];
            if ($attempt == 0) {

                $this->session-
>set userdata('login attempt',1);
                return false;

            } else if ($attempt > 0) {

                $attempt = $this->session-
>userdata['login attempt'] + 1;
                $this->session-
>set_userdata('login_attempt',$attempt);
```

Gambar 4.7 Rancang Kode Model Login

4.4.2 Implementasi Halaman Utama Beranda Admin

Gambar 4.8 merupakan implementasi dari halaman beranda admin ketika admin berhasil masuk kedalam sistem. Pada halaman dashboard utama admin terdiri dari beberapa *view* yaitu *header, body, dan footer*.

Gambar 4.8 Halaman Dashboard Utama

4.4.2.1 Implementasi Header View

Pada view header terdapat informasi dari level admin dan nama adimn yang berhasil masuk ke dalam sistem berdasarkan klasifikasi admin menurut wilayah. Pada nama admin disebelah pojok kanan terdapat fungsi *dropdown* yang berisi tombol logout. Gambar 4.9 merupakan header view.

Gambar 4.9 Header View

Karena terdapat beberapa penggolongan admin maka ada beberapa fungsi yang dijalankan pada controller untuk menampilkan level serta nama admin.

4.4.2.1.1 Implementasi Tombol Logout Pada Header

Fungsi dari tombol logout adalah untuk mengeluarkan admin dari sistem yaitu dengan cara menghancurkan dari sesi admin yang telah dismpan dalam database kemudian sistem akan mengupdate aktivitas yang sebelumnya *logging-in* mempunyai status *yes* berubah status menjadi *no*. Gambar 4.10 Merupakan implementasi tombol logout pada header.

```
        $token = $this->session->userdata('token');
        $logged = "no";
        $this->load->model('admin model');
        $this->login model-
>update_users_activity($token,$logged);
                $this->session->sess_destroy();
        redirect(base_url().'admin');
    }
```

Gambar 4.10 Implementasi tombol logout

4.4.2.2 Implementasi *Sidebar Page Body*

Sidebar utama merupakan badan halaman berbentuk sidebar yang berisi beberapa tombol yang mengarahkan kepada fungsionalitas admin, namun pada bagian level id TPS ada beberapa fungsionalitas yang tidak sama dengan yang lain, karena

mempunyai hak akses yan berbeda. Gambar 4.11 merupakan implementasi dari badan halaman utama yang berbentuk sidebar.

Gambar 4.11 Sidebar utama pada body

4.4.2.2.1 Tombol Dashboard

Tombol Dashboard merupakan tombol yang mengarahkan kepada halaman dashboard admin yang berisi div informasi dari daerah admin dan beberapa tombol yang berfungsi untuk memvalidasi data pemilih dan tombol untuk membuka halaman rekap partai. Gambar 4.12 merupakan tampilan dari halaman dashboard admin

Gambar 4.12 Tampilan Dashbord

Ada beberapa perbedaan untuk memastikan bahwa halaman itu merupakan halaman admin berdasarkan level maka harus ditampilkan wilayah admin tersebut untuk itu ada beberpa controller maupun model yang berupa fungsi implementasi dari berbagai level admin.

4.4.2.2.1.1 Tombol Validasi Pemilih

Tombol validasi pemilih mengarahakan kepada halaman validasi pemilih dimana admin mempunyai wewenang untuk memvalidasi pemilih yaitu memastikan bahwa pemilih masih aktif artinya pemilih tidak sedang pindah atau meninggal. Gambar 4.13 menunjukkan implementasi halaman validasi pemilih tabel menggunakan plugin datatables sehingga admin dapat memfilter data pemilih.

Gambar 4.13 Halaman Validasi Pemilih

Gambar 4.14 merupakan rancang kode dari impelementasi halaman validasi pemilih.

```
$(document).ready(function(){
    $('#tabeldata').dataTable();
    $('#select-all').click(function(){
        if(this.checked){
            $(':checkbox').each(function(){
                this.checked = true;
            });
        } else{
            $(':checkbox').each(function(){
                this.checked = false;
```

```
                });
            }
        });
        $('#select-all2').click(function(){
            if(this.checked){
                $(':checkbox').each(function(){
                    this.checked = true;
                });
            } else{
                $(':checkbox').each(function(){
                    this.checked = false;
```

Gambar 4.14 Rancang Kode Validasi Pemilih

Ada beberapa tombol pada halaman validasi pemilih.

4.4.2.2.1.1.1 Tombol Ubah Validasi pada Tabel Menu

Merubah status validasi setiap satu data pemilih pada tabel, berikut kode implementasi dari ubah validasi. Gambar 4.15 menunjukkan sumber kode dari tombol ubah validasi.

```
function set_validasi($id)
    {
        $statusOk ="1";
        $statusNo ="0";

        $hasil = $this->admin model-
>cek validasi pemilih($id);
        if($hasil == 1){
        $data = array(
                'Status'                => $statusNo
            );
        }
        elseif($hasil == 0){
        $data = array(
                'Status'                => $statusOk
            );
        $this->admin_model->update_data_pemilih($id,$data);
```

Gambar 4.15 Rancang Kode Ubah validasi

4.4.2.2.1.1.2 Tombol Ubah *All Selected*

Tombol untuk merubah validai dari semua baris pada tabel yang dicentang dengan texbox. Tombol ini hanya merubah

validasi menjadi ok dan tidak sebaliknya. merupakan menunjukkan rancang kode dari implementasi ubah *all selected* Gambar 4.16.

```
function set_all_validasi()
        $statusOk ="1";
        $nik = array();
        $nik = $this->input->post('checkbox');
        $data = array(
                'Status'                => $statusOk
        $this->admin_model-
>update all data pemilih($nik,$data);
        $this->session->set_flashdata
```

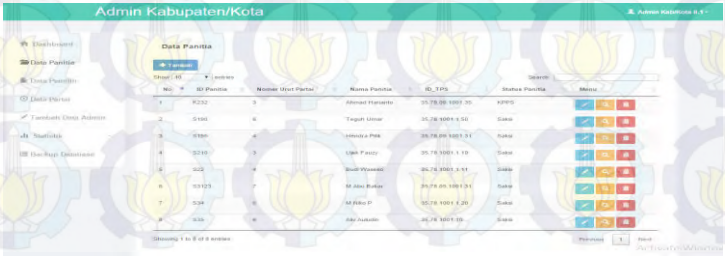
Gambar 4.16 Rancang Kode Ubah *Allselected*

4.4.2.2.1.2 Tombol Rekap Partai

Tombol rekap partai yang berada pada halaman dashboard merupakan tombol yang mengarahkan pada fungsi rekap partai yang mengikuti pemilihan. Halaman ini juga dideskripsikan pada tombol fungsionalitas pada sidebar.

4.4.2.2.2 Tombol Data Panitia

Tombol data pemilih mengarahkan pada halaman yang berisi data semua panitia yang berhasil disaring sesuai dengan level admin fungsi dari halaman. untuk mengolah data panitia baik itu menambah merubah dan menghapus data panitia, sama seperti pada tabel sebelumnya tabel, tabel data panitia menggunakan plugin datatable untuk memudahkan pencarian dan penyortiran data seperti yang diberikan pada Gambar 4.17.

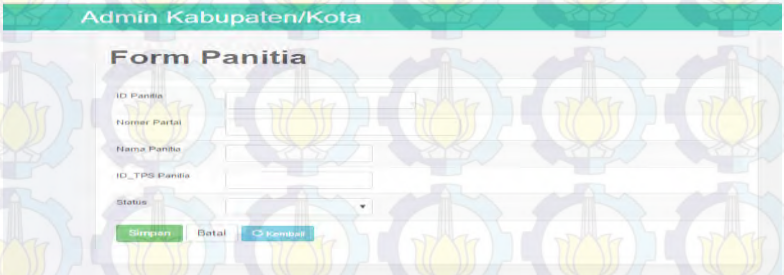


Gambar 4.17 Halaman Data Panitia

Halaman data panitia ini memilik beberapa tombol untuk melakukan pengolahan yaitu tombol tambah, edit, detail dan hapus data panitia.

4.4.2.2.2.1 Tombol Tambah atau Edit

Pada dasarnya tombil ini mengarahkan pada halaman yang sama yaitu form panitia, hanya saja ada uri yang dibedakan, jika uri adalah tambah maka secara otomatis text box pada halaman kosong dan jika uri adalah edit maka text box pada form panitia mempunyai nilai sesuai dengan data panitia yang diedit berdasarkan no Idpanitia**.** Gambar 4.18 merupakan contoh form panitia.



Gambar 4.18 Form Panitia

Pada halaman tersebut terdaat 3 tombol yaitu, tombol batal yaitu menjadikan textbox semua mempunyai nilai nol, tombol kembali mengarahkan ke halaman sebelumnya dan tombol simpan mempunyai fungsi menyimpan data yang telah diisi pada form. Gambar 4.19 merupakan contoh dari rancang kode dari tambah maupun edit panitia.

```
if ($mau_ke == "add") {//jika uri segmentnya add
    $data['title'] = 'Tambah panitia';
    $data['aksi'] = 'aksi add';
    $this->load->view('include/header',$data);
    $this->load->view('kota/view_addPanitia',$data);
    $this->load->view('include/footer');
```

```
        } else if ($mau_ke == "edit") {//jika uri segmentnya
} else if ($mau_ke == "aksi_add") {//jika uri segmentnya
aksi add sebagai fungsi untuk insert
            $data = array(
                'ID_PANITIA'          => $id,
                'NMR_URUT_PARTAI'     => $nomer,
                'NAMA_PANITIA'        => $nama,
                'ID_TPS'              => $id_tps,
                'STATUS_PANITIA'      => $status,
            );
            $this->admin_model->insert_data_panitia($data);
//model insert
```

Gambar 4.19 Rancang Kode Form Panitia

4.4.2.2.2.2 Tombol Detail Panitita

Tombol detail mempunyai fungsi untuk menampilkan data panitia yang tidak ada pada tabel. Seperti status persetujuan panitia terhadap hasil pemilihan. Gambar 4.20 merupakan contoh dari tampilan halaman detail panitia.

Gambar 4.20 Halaman Detail Panitia

Berikut adalah kode sumber dari detail panitia yang dipilih berdasarkan nomer ID panitia yang ditunjukkan pada Gambar 4.21.

```
public function detail_panitia($id){ //fungsi detail Panitia
        $this->cek_admin();
        $data['admin'] = $this->get_namaadmin();
```

```
        $data['level']=$this->get leveladmin();
        $data['title'] = 'Detail Panitia';
        $data['qdata'] = $this->admin model-
>get data panitia($id); //query model barang sesuai id
        $this->load->view('include/header',$data);
        $this->load->view('kota/view_detPanitia',$data);
//meload views detail barang
        $this->load->view('include/footer');
```

Gambar 4.21 Kode Sumber Detail Panitia

4.4.2.2.2.3 Tombol Hapus Panitia

Fungsi tombol hapus panitia adalah untuk menghapus data panitia berdasarkan nomer ID panitia. Gambar 4.22merupakan kode sumber hapus panitia.

```
 public function hapus($gid){
        $this->admin_model->delete_data_panitia($gid);
        $this->session->set_flashdata("pesan", "<div
class=\"alert alert-danger\" id=\"alert\"><i
class=\"glyphicon glyphicon-ok\"></i> Data berhasil
dihapus</div>");
        redirect('Admin_kota/get_panitia');
    }
```

Gambar 4.22 Kode Sumber Hapus panitia

4.4.2.2.3 Tombol Data Pemilih

Tombol Data Pemilih pada sidebar berfungsi untuk megarahkan pada halaman yang berisi tabel data pemilih, pada data pemilih ini kewenangan admin hanya untuk mengedit atau menambah data pemilih. Data panitia yang ditampilkan pada tabel sudah disaring menurut level wilayah admin, dan untuk memudahkan pencarian, tabel menguunakan *plugin* datatables yang memudahkan untuk melakukan penyaringan dengan memasukkan kata kunci pada sisi pojok kanan atas tabel serta dapat mengurutkan tabel menurut atribut tabel tersebut. Gambar 4.23 merupakan implementasi dari halaman pemilih.

Gambar 4.23 Halaman Data Pemilih

4.4.2.2.3.1 Tambah dan Edit Data Panitia(form panitia)

Kedua tombol ini mengarah pada halaman yang sama tapi mempunyai uri yang berbeda untuk menampilkan halaman, ketika uri adalah add maka textbox pada form kosong, jika uri adalah edit maka texbox pada form bernilai data dari id pemilih yang ingin diubah. Gambar 4.24 merupakan bentuk tampilan dari form pemilih.

Gambar 4.24 Halaman Form Pemilih

Ada beberapa tombol yang ditampilkan pada halaman form pemilih, yaitu tombol batal yang berfungsi untuk mereset nilai yang ada pada text box, kembali berfungsi untuk mengembalikan pada halaman sebelumnya dan simpan

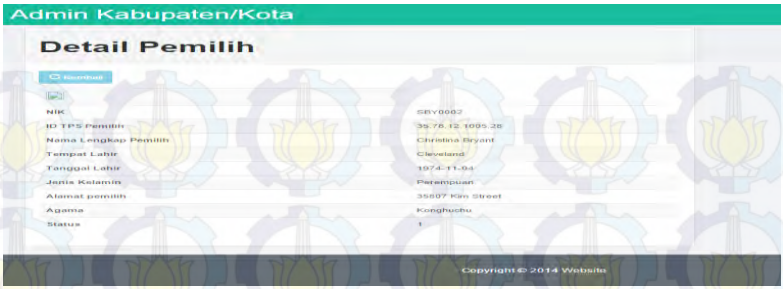
menambahkan data kedalam database menurut form yang telah diisi. Gambar 4.25 Merupakan kode sumber dari fungsi tombol simpan.

```
        if ($mau_ke == "add") {//jika uri segmentnya add
            $data['title'] = 'Tambah panitia';
            $data['aksi'] = 'aksi add';
            $this->load->view('include/header',$data);
            $this->load->view('kota/view_addPemilih',$data);
            $this->load->view('include/footer');
        } else if ($mau_ke == "edit") {//jika uri segmentnya
edit
            $data['qdata']  = $this->admin model-
>get_data_pemilih($idu);
            $data['title'] = 'Edit';
            $data['aksi'] = 'aksi_edit';
            $this->load->view('include/header',$data);
            $this->load->view('kota/view addPemilih',$data);
            $this->load->view('include/footer');
        } else if ($mau_ke == "aksi_add") {//jika uri
segmentnya aksi_add sebagai fungsi untuk insert
            $data = array(
                'NIK'                 => $nik,
                'ID TPS'              => $id tps,
                'NAMA LENGKAP'        => $nama,
                'TEMPAT_LAHIR'        => $tempat_lahir,
                'TANGGAL_LAHIR'       => $tanggal_lahir,
                'JENIS_KELAMIN'       => $jenis_kelamin,
                'ALAMAT_LENGKAP'      => $alamat,
                'AGAMA'               => $agama,
                'FOTO'                => $this->upload-
>do_upload(),
Save to databse ($data)
            );
```

Gambar 4.25 Rancang Kode Form Tambah/edit Pemilih

4.4.2.2.3.2 Tombol Detail

Tombol detai mempunyai fungsi untuk menampilkan data masing – masing pemilih secara detail pada halaman pemilih. Gambar 4.26 merupakan contoh dari halaman detail pemilih.



Gambar 4.26 Halaman Detail Pemilih

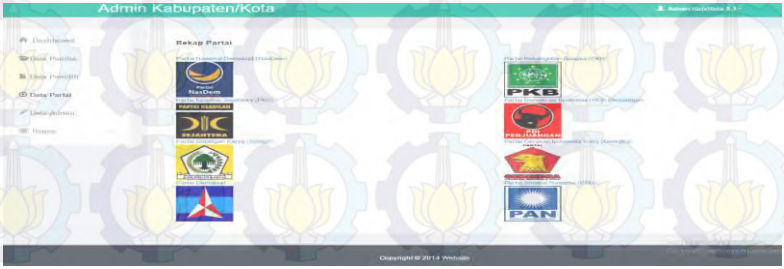
Gambar 4.27 merupakan kode sumber dari detail pemilih.

```
function detail_pemilih($id){
        $this->cek_admin();
        $data['admin'] = $this->get_namaadmin();
        $data['level']=$this->get_leveladmin();
        $data['title'] = 'Detail Panitia';
        $data['qdata'] = $this->admin model-
>get data pemilih($id);
        $this->load->view('include/header',$data);
        $this->load->view('kota/view_detPemilih',$data);
        $this->load->view('include/footer');}
```

Gambar 4.27 Rancang Kode Detail Pemilih

4.4.2.2.4 Tombol Data Partai

Tombol Data partai mempunyai fungsi mengarahkan pada halaman rekap partai dimana pada halaman itu ditampilkan partai – partai yang terdaftar pada pemilu. Gambar 4.28 merupakan contoh dari halaman rekap partai.



Gambar 4.28 Halaman Data Partai

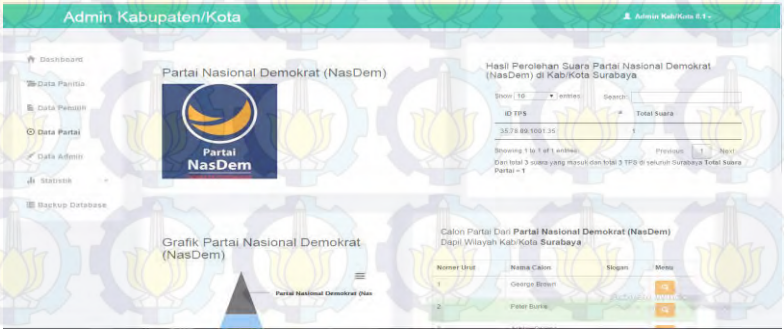
Pada setiap gambar partai mengarahkan pada halaman detail dari data partai tersebut. Gambar 4.29 merupakan kode sumber dari tombol Data partai dan halaman partai.

```
function rekapPartai()
    {
        $this->cek admin();
        $data['admin'] = $this->get namaadmin();
        $data['level']=$this->get_leveladmin();
        $data['partai'] = $this->admin_model->get_partai();
        $id  = $this->get_idadmin();
        $this->tempid  = substr($id,2,-3);
        $data['hasil'] = $this->admin model-
>get_suara($this->tempid);
        $this->load->view('include/header',$data);
        $this->load->view('kota/view_rekapPartai',$data);
        $this->load->view('include/footer');
    }
```

Gambar 4.29 Rancang Kode Data Partai

4.4.2.2.4.1 Detail Partai

Tombol Detail partai ditekan pada masing-masing gambar partai maka akan mengarahkan ke halaman partai yang ditekan, Pada halaman tersebut terdapat empat bagian(div) div pertama logo partai, div yang kedua tabel data perolehan suara partai, div yang ketiga grafik suara partai dan div yang keempat adalah tabel dari calon partai yang disaring berdasarkan wilayah level admin. Gambar 4.30 merupakan halaman dari detail partai.



Gambar 4.30 Halaman Detail Partai

Gambar 4.31 adalah kode sumber dari halaman detail partai.

```
$(function () {
    $('#chart').highcharts({
        chart: {
            type: 'pyramid',
            marginRight: 100
        },
        title: {
            text: '',
            x: -50
        },
        plotOptions: {
            series: {
                dataLabels: {
                    enabled: true,
                    format: '<b>{point.name}</b>
({point.y:,.0f})',
                    color: (Highcharts.theme &&
Highcharts.theme.contrastTextColor) || 'black',
                    softConnector: true
                }
            }
        },
        legend: {
            enabled: false
        },
        series: [{
            name: 'Total Suara',
            data: [
                ['Data Suara Masuk']
            ]
        }]
    });
});
```

Gambar 4.31 Rancang Kode Detail Partai

4.4.2.2.4.2 Tombol Detail Calon Partai

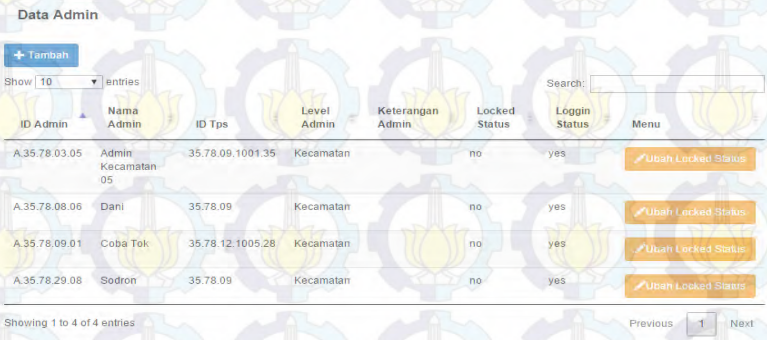
Tombol detail calon terletak pada halaman detail partai, tombol detal calon mempunyai fungsi mengarahkan pada halaman detail data calon yang berisi 3 bagian div, div yang pertama merupakan data calon, div yang kedua hasil perolehan suara calon dan dihitung dari beberapa level wilayah, serta grafik

yang ketiga adalah data perolehan berupa grafik. Gambar 4.32 merupakan halaman detail calon.

Gambar 4.32 Detail Calon Partai

4.4.2.2.5 Tombol Data Admin

Tombol Data admin mengarahkan pada halaman data admin yanag berbebentuk tabel, fungsi halaman ini yaitu untuk merubah *set lock* admin dan menambah data admin sepertiyang ditunjukkan pada gambar 4.34



Gambar 4.33 Halaman Data Admin

4.4.2.2.5.1 Tombol Ubah *locked* Status

Tombol ini berfungsi untuk merubah status admin yang terkunci karena gagal masuk sebanyak tiga kali pada Gambar 4.34 merupakan kode untuk merubah status lock pada admin.

```
$lockyes = "yes";
        $lockno = "no";
        $aksi = "Ubah Lock Status Admin";
         $hasil = $this->admin_model->cek_lock_admin($id);
         if($hasil == yes){
         $data = array(
                 'locked_status'                    => $lockno
             );
         }
         elseif($hasil == 0){
         $data = array(
                 'locked_status'                    => $lockyes
             );
```

Gambar 4.34 Rancang Kode Ubah *Lock* Status

4.4.2.2.5.2 Tombol Tambah Admin

Tombol Tambah Admin mempunyai fungsi untuk menambah admin, beberapa golongan admin memperoleh hak akses untuk menambah admin dari golongan bawahnya, misalnya level admin Kecamatan hanya boleh menambah admin level kelurahan. Gambar 4.35 Merupakan form dari tambah admin, id admin otomatis *generate* ketika admin memilih daerahnya.

Gambar 4.35 Form Admin

4.4.2.2.6 Implementasi Tombol Backup Database

Tombol *backup* database mempunyai fungsi untuk *memback up* database yang dapat diunduh oleh admin dalam format .rar, adapun untuk memudahkan klarifikasian penamaan, backup database berdasarkan nama admin dan tanggal saat database di*backup.* untuk kemanan . Gambar 4.36 merupakan kode sumber backup database.

```
        $this->load->dbutil();
        $nama = $this->get namaadmin();
        $prefs = array(
                'tables'      => array('panitia',
'partai','pemilih','temp_suara_partai','temp_suara_partai'),
                 'format'      => 'zip',
                 'filename'    => 'my db backup.sql'
               );
        $backup =& $this->dbutil->backup($prefs);

        $db_name = $nama.'backup-on-'. date("Y-m-d-H-i-s")
.'.zip';
        $save = 'pathtobkfolder/'.$db name;

        $this->load->helper('file');
        write_file($save, $backup);
        $this->load->helper('download');
        force_download($db_name, $backup)
```

Gambar 4.36 Kode Sumber Dari *Backup* Data

4.5 Implementasi Halaman Admin KPU

Pada halaman admin KPU ini hanya ada satu fungsi utama yaitu membuka data suara, untuk membuka data suara ini memerlukan *private key* untuk mendekrip data suara yang terenkripsi menggunakan algoritma RSA, Berikut merupakan implementasi dari proses pembukaan suara oleh Admin KPU yang ditunjukkan pada Gambar 4.37.

```
enkripsi($suara)
  {

      $keypriv ="";
         $rsa = new Crypt_RSA();
         $rsa-
>loadKey($keypriv,CRYPT_RSA_PRIVATE_FORMAT_XML);
         $rsa->setEncryptionMode(CRYPT_RSA_ENCRYPTION_NONE);
         $wesdec = base64_decode($data);
         return $hasil =  trim($rsa->decrypt($wesdec));
  }
  public function buka_suara()
  {
    $no = 0;
   $data = get_all_suara();
   foreach ($data ) {
      $no++;
       $tot = 1;
       $TPS = $key->NMR_ID_TPS ;
      $has_calon  = enkripsi($key->ENKR_PILIHAN_CALON);
       $has_partai = enkripsi($key->ENKR_PILIHAN_PARTAI);
       $hasil = array(
                        'NO' => $no,
                        'ID_TPS' =>$TPS,
                         'PARTAI'=>$has_partai,
                         'CALON'=> $has_calon,
                         'TOTAL' => $tot,
       );
       insert_to_database($hasil);
      }
```

Gambar 4.37 Rancang Kode Buka Suara

BAB V
UJI COBA DAN EVALUASI

Bab uji coba dan evaluasi berisi mengenai hasil uji coba dan evaluasi terhadap aplikasi web *e-voting*.

5.1 Lingkungan Uji Coba

Lingkungan uji coba yang digunakan dalam pembuatan Tugas Akhir ini meliputi perangkat lunak dan perangkat keras yang digunakan untuk melakukan uji coba web aplikasi *e-voting.* Lingkungan uji coba merupakan komputer tempat uji coba sistem yang spesifikasinya dijelaskan sebagai berikut:

1. Perangkat Keras
   a. Intel(R) Core(TM)i5-2310M @ 1,8GHz
   b. Memori 4,00 GB
   c. Tipe Sistem Operasi 64-bit
2. Perangkat Lunak
   a. Sistem Operasi Microsoft Windows 8 Home
   b. Pustaka *dataTables, highchart, codeigniter.*
   c. Perangkat tambahan: Web Server IIS 8.5, dan XAMPP yang meliputi PHPMyAdmin dan sistem manajemen basis data.

5.2 Skenario Uji Coba

Pada bagian ini dijelaskan mengenai skenario uji coba yang telah dilakukan. Ada dua jenis uji coba yang telah dilakukan, yaitu:

1. Uji coba fungsionalitas

   Jenis uji coba ini berfungsi untuk menguji fungsionalitas dari aplikasi sistem yang dibuat. Uji coba yang telah dilakukan, yaitu:

   a. Login berdasarkan level Admin
   b. Mengolah menghapus data Panitia
   c. Mengolah data Pemilih

d. Menambah admin sesuai level admin

e. Menyimpan aktivitas admin dalam data *session*

f. Menampilkan suara partai dan calon dalam bentuk tabel dan grafik.

g. Backup database setiap level admin

2. Uji coba keamanan

Jenis uji coba ini berfungsi untuk menguji keamanan dari aplikasi sistem yang dibuat. Uji coba yang telah dilakukan adalah:

a. Percobaaan login

b. Masuk kedalam admin level lain.

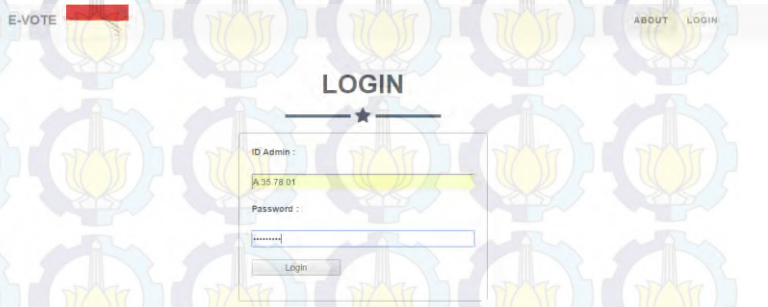
c. Pengujian dengan perangkat lunak acunetix

5.2.1 Uji Coba Fungsionalitas

5.2.1.1 Skenario Pengujian 1: *Login Admin*

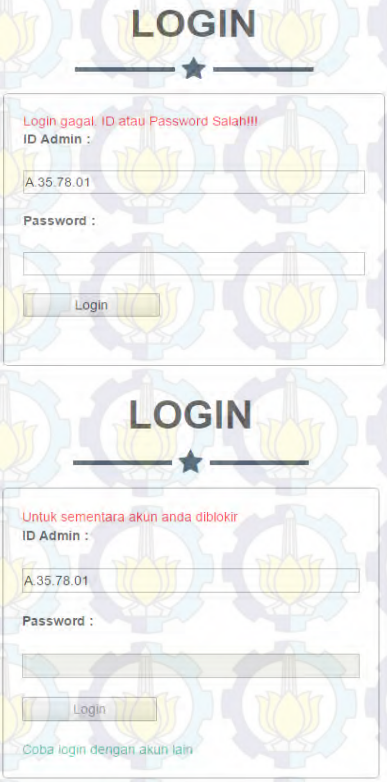
Pada skenario ini pengujian dilakukan untuk melakukan pengujian login menurut admin level.

Gambar 5.1 merupakan form login untuk admin yang akan melakukan login



Gambar 5.1 Form login

Jika login gagal akan ada tanda pada form bahwa ID atau Password salah, Dan jika admin mencoba login sebanyak 3 kali maka secara langsung id admin akan diblokir. Dan pada *textbox* password akan di mode *disabled.* Gambar 5.2 merupakan contoh ketika admin tidak berhasil masuk ke dalam sistem.



Gambar 5.2 Pemberitahuan login gagal

5.2.1.1.1 Login Sukses Admin level kabupaten/kota

Halaman ketika admin kota sukses melakukan login, secara langsung admin kabupaten/kota akan diarahkan pada halaman dashboard admin yang ditunjukkan pada Gambar 5.3.

Gambar 5.3 Halaman Admin kota

5.2.1.1.2 Login Sukses Admin Kecamatan

Halaman ketika admin kota sukses melakukan login, secara langsung admin kecamatan akan diarahkan pada halaman dashboard admin kecamatan yang ditunjukkan pada Gambar 5.4.

Gambar 5.4 Halaman Admin kecamatan

5.2.1.1.3 Login Sukses Admin level Desa/kelurahan

Halaman ketika admin kota sukses melakukan login, secara langsung admin Desa/Kelurahan akan diarahkan pada halaman dashboard admin yang ditunjukkan pada Gambar 5.5.

Gambar 5.5 Halaman Desa/kelurahan

5.2.1.1.4 Login Sukses Admin level TPS

Halaman ketika admin TPS sukses melakukan login, secara langsung admin kota akan diarahkan pada halaman dashboard admin TPS yang ditunjukkan pada Gambar 5.6.

Gambar 5.6 Halaman Admin TPS

5.2.1.2 **Skenario Pengujian 2: Mengolah data panitia**

Dalam skenario ini, dilakukan pengolahan data panitia semua level admin mempunyai wewenang untuk ini kecuali admin TPS yang hanya bisa melihat panitia pada wilayahnya,

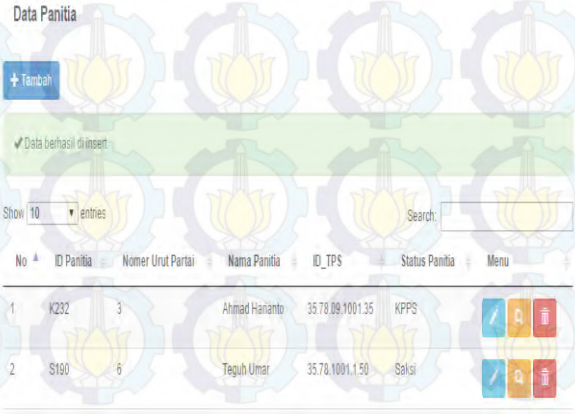
pengolahan ini berupa menambah, merubah dan menghapus data panitia.

5.2.1.2.1 Menambah data panitia

Admin masuk ke dalam sistem kemudaian admin memilih tombol tambah panitia dan setelah admin mengisi form panitia seperti pada Gambar 5.7

Gambar 5.7 Form Panitia

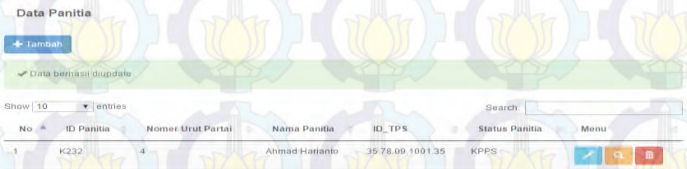
Jika berhasil maka akan kembali ke tabel data panitia dan muncul tanda seperti pada Gambar 5.8.



Gambar 5.8 Berhasil Menambah

5.2.1.2.2 **Merubah data panitia**

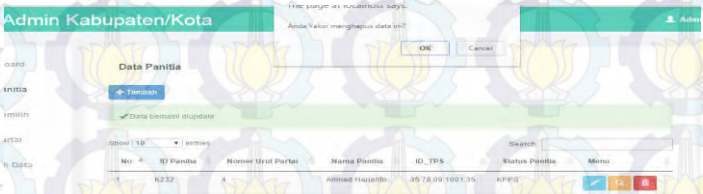
Admin memilih tombol edit pada bagian samping tabel pantia maka secara langsung akan mengarahkan pada halaman form dan uri yang bersi idpanitia seperti pada Gambar 5.7. Maka ada pemberitahaun jika data berhasil dirubah seperti yang ditunjukkan pada Gambar 5.9.



Gambar 5.9 Data Berhasil Diubah

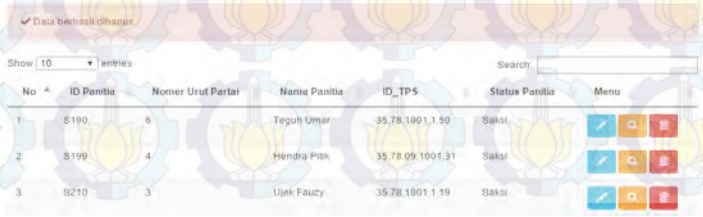
5.2.1.2.3 **Menghapus data panitia**

Admin masuk ke halaman data panitia dan tekan tombol hapus yang terletak pada bagian menu yang terdapat pada tabel Maka akan ada a*lert* seperti pada Gambar 5.10**.**



Gambar 5.10 *alert* Saat pada hapus

Jika hapus sukses maka akan ada pemberitahuan, seperti pada Gambar 5.11



Gambar 5.11 *Alert* setelah dihapus

5.2.1.3 Skenario Pengujian 3: Mengolah data Pemilih

Pada skenario ini admin hanya dapat mengolah dengan menambah dan merubah data pemilih.

5.2.1.3.1 Menambah data pemilih

Admin masuk ke dalam sistem kemudian, setelah admin mengisi form panitia seperti pada Gambar 5.12

Gambar 5.12 Form Pemilih

Jika berhasil maka akan kembali ke tabel data panitia dan muncul tanda seperti pada Gambar 5.13

Gambar 5.13 *alert* setelah data fitambah

5.2.1.3.2 Merubah data pemilih

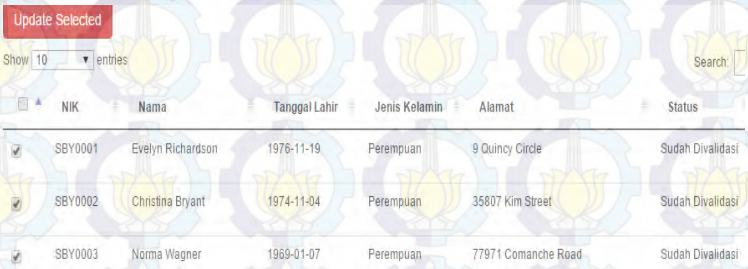
Admin memilih tombol edit pada bagian samping tabel pantia maka secara langsung akan mengarahkan pada halaman form dan uri yang bersi idpanitia seperti pada Gambar 5.12.

Maka ada pemberitahaun jika data berhasil dirubah seperti Gambar 5.14**.**

Gambar 5.14 *alert* setelah data di ubah

5.2.1.3.3 Memvalidasi Pemilih

Admin masuk kedalam halaman Validasi pemilih , kemudian admin memilih dengan *check_box* pada halaman ini terdapat fitur untuk menu pilih semua sehingga memudahkan admin untuk merubah beberapa validasi dari data pemilih contoh *check_box* seperti pada Gambar 5.15



Gambar 5.15 Tabel dan Checkbox

Jika sukses divalidasi maka akan muncul pemberitahuan seperti pada Gambar 5.16

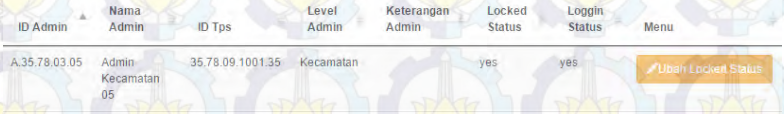
Gambar 5.16 Setelah berhasil diubah

5.2.1.4 Skenario Pengujian 4 Mengolah data admin

Pada skenario ini admin dapat mengolah admin daerah level bawahnya. Namun dalam pengelolaan admin hanya bisa menambah data admin pada level bawahnya dan merubah *lock* status dari admin yang telah gagal login dan akun terkunci. Ada fungsi untuk mengaktfkanya kembali yaitu pada tombol ubah *lock* status admin.

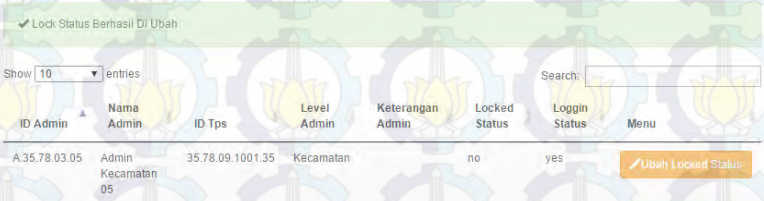
5.2.1.4.1 Merubah status *locked* admin

Pada skenario ini aakan dilakukan perubahan status admin yang telah terkunci dikarenakan gagal login admin yang berhak melakukan hal ini adalah admin yang berada pada level tingkat dareah diatasnya, admin menekan tombol ubah status *lock* seperti pada Gambar 5.17.



Gambar 5.17 Tombol ubah status *lock*

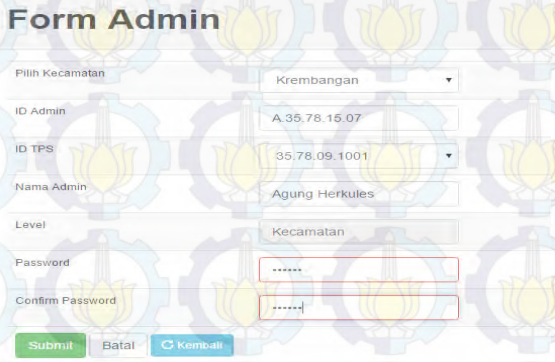
Maka setelah ditekan tombol tersebut status akan ada pemberitahuan seperti pada Gambar 5.18 dan status berubah menjadi *no* dan admin mencoba untuk masuk kedalam sistem.



Gambar 5.18 Lock Status setelah diubah

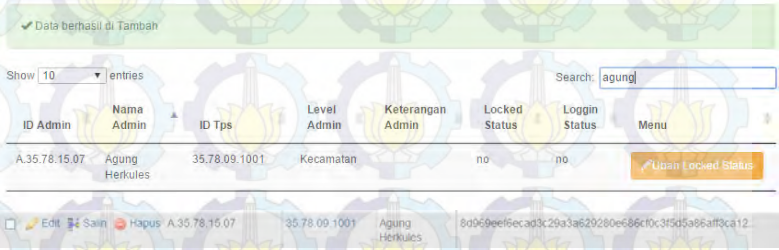
5.2.1.4.2 **Menambah Admin**

Pada sekenario ini akan dilakukan penambahan admin, admin hanya dapat menambah pada level dibawahnya, pada form disediakan *dropdown options* untuk memilih daerah level dibawah admin, kemudian secara acak ID akan otomatis terisi dengan *string* acak seperti pada Gambar 5.19**.**



Gambar 5.19 From Admin

Setelah berhasil diisisi maka akan kembali langsung pada halaman admin dan data tersimpan pada database. Seperti yang terlihat pada Gambar 5.20



Gambar 5.20 Hasil setelah admin berhasil ditambah

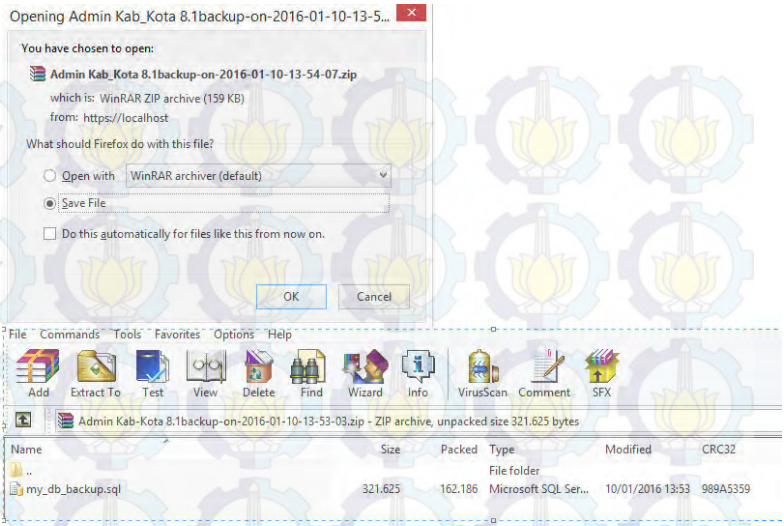
5.2.1.5 Skenario Pengujian 5: Menampilkan data suara dalam bentuk tabel dan grafik.

Dalam skenario, dilakukan uji coba untuk menampilkan hasil pemilihan dalam bentuk tabel dan grafik menurut level daerah. Pemilih memilih masuk ke dalam halaman statistik partai kemudian tekan tombol *show* grafik di bawah tabel tersebut. Berikut merupakan hasil pemilihan umum di daerah Surabaya yang ditunjukkan pada Gambar 5.21

Gambar 5.21 Grafik Suara Hasil Pemilu

5.2.1.6 Skenario Pengujian 6: *Backup* Database

Dalam skenario ini, dilakukan apakah admin berhasil mmbackup database, hal pertama yang dilakukan admin adalah menekan tombol *Backup Database* kemudian secara langsung akan mengunduh data berupa file yang mempunyai format *.zip* dan bernama admin yang mengunduh serta waktu unduh, folder *.zip* tersebut berisi file *my_backup_db.sql* seperti yang ditunjukkan pada Gambar 5.22

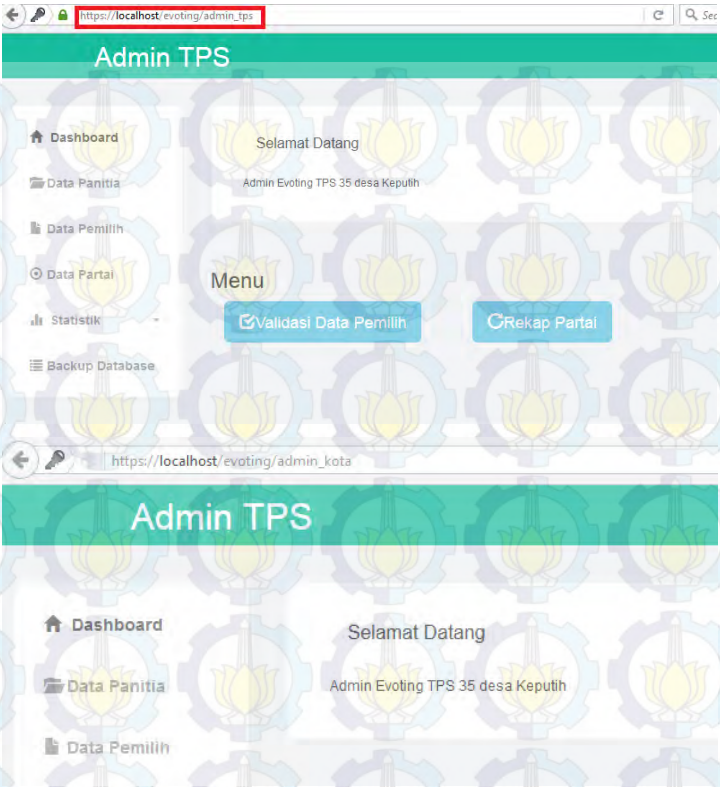


Gambar 5.22 *Backup* Database

5.2.2 Uji Coba Keamanan

5.2.2.1 Skenario Pengujian 1: Masuk Ke halaman Level lain

Dalam skenario ini, dilakukan uji coba untuk masuk ke halaman level lain saat admin sudah berhasil login, Misalkan ada admin level TPS melakukan login dan berhasil login dengan akun level TPS, maka secara otomatis admin sudah mendapatkan *session* login, maka untuk mengakses halaman lain kemungkinan bisa dilakukan dengan merubah uri, tapi dengan adanya penerapan simpan data *session* pada setiap awal login menggunakan data user admin maka hal itu dapat di minimalisir, percobaan dapat dilihat seperti pada Gambar 5.23

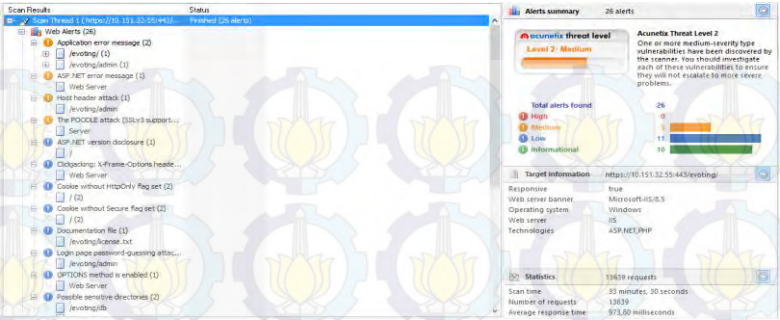


Gambar 5.23 Perubahan URI

Ketika dijalankan maka secara langsung halaman akan dikembalikan ke halaman login dan *session* data dari admin tersebut akan dihancurkan.

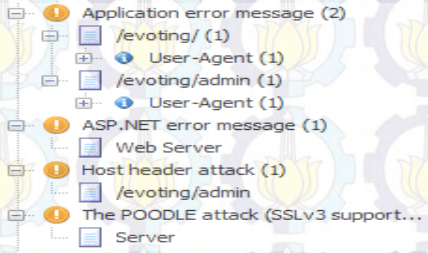
5.2.2.2 Skenario Pengujian 2: Pengujian dengan Perangkat Lunak Acunetix

Dalam skenario ini, dilakukan uji coba untuk mengetahui kerentanan dari web aplikasi *e-voting* dengan perangkat lunak acunetix. Dari pegujian tersebut ada beberapa kerentanan pada aplikasi web *e-voting* yang ditunjukkan pada Gambar 5.24.



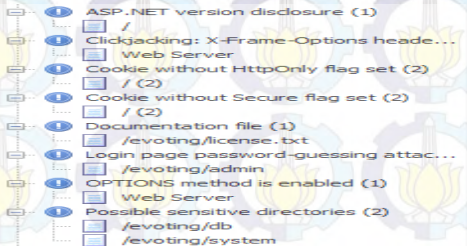
Gambar 5.24Hasil Pengujian dengan Acunetix

Terdapat beberapa uraian hasil pengujian yang dilakukan oleh perangkat lunak acunetix, baik *threat* bersifat *medium* atau *low*, Gambar 5.25 merupakan hasil kerentanan dengan level medium.



Gambar 5.25 *Medium Threat*

Sedangkan pada uraian hasil pengujian dengan level *threat low* ditunjukkan pada Gambar 5.26.



Gambar 5.26 *Low Threat*

Masing – masing *threat* yang ditemukan dalam proses pengujian akan diuraikan mengenai letak file yang kemungkinan rentan terhadap *threat* dan keterangan *threat* pada Tabel 5.1

Tabel 5.1 Tabel Hasil Pengujian web *e-voting* oleh acunetix

Jenis Kerentantan	**Ditemukan pada**	**Keterangan**	**Level**
Appplication error message	/Evoting/	Aplikasi menggeluarkan kode error, memungkinkan penyerang untuk masuk kedalam sistem	Medium
ASP.net error message	/web server	Error pada aplikasi .net	Medium
Host Header attack	/evoting	Seorang penyerang dapat memanipulasi Host header seperti yang terlihat oleh aplikasi web	Medium
The Poodle Attack	/Web Server	Seorang penyerang mungkin dapat memanfaatkan masalah ini untuk melakukan man-in-the-middle	Medium
ASP.net disclosure	/	Header HTTP dapat memberikaninformasi sensitif.	Low
Click Jacking X-frame Oprions	/server	Dampaknya dapat terjadi pada aplikasi	Low

Jenis Kerentanta n	Ditemuk an pada	Keterangan	Level
Cookie without HttpOnly flag set	/	Tidak ada impact yang sensitif terhadap aplikasi	Low
Documentati on file	/evoting/l ecensie.te xt	Mungkin penyerang dapat menemukan informasi dari dokumen tersebut	low
Login Page password guessing attack	/evote/ad min	Penyerang dapatmelakukanbanyak percobaan login	Low
Options Method is enable	/web server	Dapat mengekspos informasi sensitif yang dapat membantu penyerang	Low
Possible sensitive directories	/evoting/d b /evotingsy stem	Mungkin penyerang dapat menemukan informasi dari dokumen tersebut	Low

Uraian hasil pengujian dengan acunetix dari setiap halaman yang terdapat pada aplikasi web e-voting dengan parameter uji yaitu pengujian dengan *Blind_SQL_Injection*, *XSS*, *CSRF* dan *SQL_Injection* dari hasil pengujian tersebut terlihat bahwa hasil serangan tersebut tidak ditemukan pada setiap halaman level admin, hasil dapat dilihat pada Tabel 5.2.

Tabel 5.2 Hasil Pengujian Setiap Halaman Web Admin

Halaman	***CSRF***	***XSS***	***SQL_Injection***	***Warning Alert***
/evoting/admin_KPU/	*Not Found*	*Not Found*	*Not Found*	*Broken Link (/evoting/css/)*
/evoting/admin_kota/	*Not Found*	*Not Found*	*Not Found*	*Not found*
/evoting/admin_kecamatan/	*Not Found*	*Not Found*	*Not Found*	*Not found*
/evoting/admin_desa/	*Not Found*	*Not Found*	*Not Found*	*Not found*
/evoting/admin_tps	*Not Found*	*Not Found*	*Not Found*	*Not found*

5.2.3 Evaluasi

Dari sejumlah skenario uji coba fungsionalitas yang telah dilakukan, kontrol-kontrol aplikasi yang dibuat dapat berjalan dengan baik. Namun ada beberapa *textbox* pada *form* yang seharusnya dibuat otomatis sehingga memudahkan pengguna untuk memasukkan data maupun mengedit data.

Dari sejumlah skenario uji coba keamanan yang telah dilakukan, Pada pengujian pertama membuktikan pembagian penggolongan level admin berjalan dengan baik. Pada pengujian kedua masi banyak terjadi kerentanan pada web server atau aplikasi. Namun kerentanan tersebut tidak bersifat krusial hanya beberapa file yang perlu diatur. Artinya dalam pengamanan yang bersifat *authentication* dan *integrity* sudah termasuk aman karena tidak ada kerentanan dari serangan *CSRF, XSS, dan SQL_Injection* pada setiap halaman admin.

BAB VI
KESIMPULAN DAN SARAN

Bab ini berisi mengenai simpulan-simpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang telah dikemukakan. Selain itu, pada bab ini terdapat juga saran yang ditujukan untuk pengembangan aplikasi lebih lanjut.

6.1 Kesimpulan

Kesimpulan yang diperoleh berdasarkan uji coba dan evaluasi yang telah dilakukan antara lain:

1. Dengan melihat hasil uji coba fungsionalitas, semua kontrol aplikasi yang dibuat dapat berjalan dengan baik.
2. Penggolongan level admin berjalan sempurna. Admin tidak bisa masuk ke level admin yang lain. Informasi hasil pemilihan umum secara online (*evoting)*, dapat ditampikan menurut daerah masing-masing level.
3. Dari sisi keamanan admin, sistem juga dapat melakukan pemantauan aktivitas admin.
4. Pengamanan bersifat *integrity* dan *authentication* berjalan dengan baik dengan melihat uji coba aplikasi menggunakan acunetix

6.2 Saran

Beberapa saran yang hendak disampaikan terkait dengan pengerjaan tugas akhir ini adalah :

1. Aplikasi dapat mengelola calon partai dalam tingkatan misalnya pemilihan legislatif tingkat nasional (DPR) dan tingkat provinsi (DPRD Provinsi).
2. Pengamanan lebih ditambahkan dalam sisi web server, dalam hal ini sistem web servis *IIS 8.5*, karena kerentanan banyak terjadi pada sisi web server.

LAMPIRAN

1. Rancangan PDM pada pengguna sebelumnya

2. Kelas Model pada aplikasi evoting (admin_model.php)

```
<?php
class Admin_model extends CI_Model {
  var $account_locked, $locked, $logged;
    function __construct()
    {
        parent::__construct();
        $this->account_locked = 'locked';
        $this->locked = 'yes';
        $this->logged = "yes";
    }
     private $table = "admin";
    private $tab_kel = "desa_kelurahan";
    private $tab_tps = "tps";
    private $nk = "kab_kota";
    private $tab_panitia = "panitia";
    private $tab_kec = "kecamatan";
    private $tab_pemilih = "pemilih";
    private $tab_partai="partai";
    private $tab_calon="calon";
    private $tab_suara="temp_suara_partai";
    private $tab_suara2="temp_suara_calon";
    private $tab_log ="log_admin";
    public $_data = array();
        //query untuk data panitia
    public function get_panitia($id)
    {
        $query = $this->db->like('ID_TPS',$id,'after')
                    ->get($this->tab_panitia);
                    if($query->num_rows()>0)
                    {

                        return $query->result();
                    }
    }

    public function get_data_panitia($id)
    {
        $this->db->from($this->tab_panitia);
        $this->db->where('ID_PANITIA',$id);

        $query = $this->db->get();
        if($query->num_rows()==1){
            return $query->result();
        }
        else
            return"panitia no found";
    }
    public function insert_data_panitia($data)
```

```
    {
        $this->db->insert($this->tab_panitia, $data);
        return true;
    }
    public function update_data_panitia($id,$data)
    {
        $this->db->where('ID_PANITIA',$id);
        $this->db->update($this->tab_panitia,$data);
        return true ;
    }
    public function delete_data_panitia($id)
    {
        $this->db->where("ID_PANITIA",$id);
        $this->db->delete($this->tab_panitia);
        if($this->db->affected_rows()==1)
        {
            return true;
        }
        return false;
    }
        // untuk data Daerah
    public function get_kab_kota($id)
    {
        $query = $this->db->where('KODE_KAB_KOTA',$id)
                 ->limit(1)->get($this->nk);
                 if($query->num_rows()){
                 $row = $query->row_array();
                 return $row['NAMA_KAB_KOTA'];
                }
                else{
                    return "NOT FOUND";
                }

                }

    public function get_kecamatan($id)
    {

        $query = $this->db->where('KODE_KECAMATAN',$id)
                 ->limit(1)->get($this->tab_kec);
                 if($query->num_rows()){
                 $row = $query->row_array();
                 return $row['NAMA_KECAMATAN'];
                }
                else{
                    return "NOT FOUND";
                }
    }
    public function get_kecamatan_kodeKota($kode_kota)
    {
```

```
        $query = $this->db-
>where('KODE_KAB_KOTA',$kode_kota)
                ->get($this->tab_kec);
                if($query->num_rows()){
                    return $query->result();
                    }
                else{
                    return "NOT FOUND";
                }
    }
    public function get_desa($id)
    {
        $query = $this->db->where('KODE_DESA_KELURAHAN',$id)
                ->limit(1)->get($this->tab_kel);
                if($query->num_rows()){
                $row = $query->row_array();
                return $row['NAMA_DESA_KELURAHAN'];
                }
                else{
                    return "NOT FOUND";
                }
    }
    public function get_desa_kodeKecamatan($kode_kecamatan)
    {

        $query = $this->db-
>where('KODE_KECAMATAN',$kode_kecamatan)
                ->get($this->tab_kel);
                if($query->num_rows()){
                    return $query->result();
                    }
                else{
                    return "NOT FOUND";
                }
    }
    public function get_TPS($id)
    {

        $query = $this->db->where('ID_TPS',$id)
                ->limit(1)->get($this->tab_tps);
                if($query->num_rows()){
                $row = $query->row_array();
                return $row['NAMA_TPS'];
                }
                else{
                    return "NOT FOUND";
                }
    }
    public function get_TPS_kodeDesa($kode_desa)
```

```
    {

        $query = $this->db->like('ID_TPS',$kode_desa)
                ->get($this->tab_tps);
                 if($query->num_rows()){
                    return $query->result();
                    }
                else{
                    return "NOT FOUND";
                }
    }

    //query untuk data pemilih
    public function get_pemilih($id)
    {
                    $query = $this->db-
>like('ID_TPS',$id,'after')
                    ->get($this->tab_pemilih);
                    if($query->num_rows()>0)
                    {

                        return $query->result();
                    }
    }
    public function get_data_pemilih($nik)
    {
        $this->db->from($this->tab_pemilih);
        $this->db->where('NIK',$nik);

        $query = $this->db->get();
        if($query->num_rows()==1){
            return $query->result();
        }
        else
            return"panitia no found";
    }
    public function insert_data_pemilih($data)
    {
        $this->db->insert($this->tab_pemilih, $data);
        return true;
    }
    public function update_data_pemilih($nik,$data)
    {
        $this->db->where('NIK',$nik);
        $this->db->update($this->tab_pemilih,$data);
        return true ;
    }
    public function update_all_data_pemilih($nik,$data)
    {
        $this->db->where_in('NIK',$nik);
```

```
        $this->db->update($this->tab pemilih,$data);
        return true ;
    }
    public function delete_data_pemilih($nik)
    {
        $this->db->where("NIK",$nik);
        $this->db->delete($this->tab_pemilih);
        if($this->db->affected_rows()==1)
        {
            return true;
        }
        return false;
    }
    public function cek_validasi_pemilih($nik)
    {
        $cek ="";
        $this->db->select('Status');
        $this->db->where("NIK",$nik);
        $this->db->limit(1);
        $query = $this->db->get($this->tab pemilih);
        if($query->row_array()>0)
        {
            foreach ($query->result() as $key) {
                $cek = $key->Status;
            }
            return $cek ;
        }
    }

    //query untuk data Partai
    public function get_partai()
    {
        $query = $this->db->get($this->tab_partai);

        return $query->result();

    }
    public function get_partai_byNo($no)
    {
        $this->db->from($this->tab partai);
        $this->db->where('NMR URUT PARTAI',$no);
        $this->db->limit(1);
        $query = $this->db->get();
        if($query->row_array()>0)
        {
            return $query->result();
        }
    }

    public function get_suara_partai($no,$id)
```

```
    {
        $this->db->from($this->tab_suara);
        $this->db->where('NMR_URUT_PARTAI',$no);
        $this->db->like('ID_TPS',$id,'after');
        $query=$this->db->get();
        if($query->row_array()>0)
        {
            return $query->result();
        }
    }
    public function get_suara($id)
    {
        $this->db-
>select('partai.NAMA_PARTAI,temp_suara_partai.TOTAL_SUARA_PA
RTAI,temp_suara_partai.ID_TPS');
        $this->db->like('ID_TPS',$id,'after');
        $this->db->from($this->tab_suara,$this->tab_partai);
        $this->db->join('partai','partai.NMR_URUT_PARTAI =
temp_suara_partai.NMR_URUT_PARTAI');
        $query = $this->db->get();
        if($query->row_array()>0)
            {

            return $query->result();
        }
    }
    public function get_SuaraCalon($id)
    {
        $this->db-
>select('calon.NAMA_CALON,temp_suara_calon.TOTAL_SUARA_CALON
,temp_suara_calon.ID_TPS');
        $this->db->like('ID_TPS',$id,'after');
        $this->db->from($this->tab_suara2,$this->tab_calon);
        $this->db->join('calon','calon.ID_CALON =
temp_suara_calon.ID_CALON');
        $query = $this->db->get();
        if ($query->row_array()>0)
        {
            return $query->result();
        }
    }
    public function get_total_suara_masuk($id)
    {
        $this->db->select_sum('TOTAL_SUARA_PARTAI');
        $this->db->like('ID_TPS',$id,'after');
        $this->db->from($this->tab_suara);
        $query = $this->db->get();

            return $query->result();
    }
```

```
    public function get_jumlah_suara_partai($no,$id)
    {
        $this->db->select sum('TOTAL SUARA PARTAI');
        $this->db->where('NMR URUT PARTAI',$no);
        $this->db->like('ID_TPS',$id,'after');
        $this->db->from($this->tab_suara);
        $query = $this->db->get();

            return $query->result();
    }
    public function get_jumlah_tps_tiapDaerah($id)

    {
        $this->db->like('ID TPS',$id,'after');
        $this->db->from($this->tab suara);
        $query = $this->db->get();

        return $query->num_rows();
    }

    //query untuk data calon

    public function get_calon_partai($no,$id)
    {
        $this->db->from($this->tab calon);
        $this->db->where('NMR URUT PARTAI',$no);
        $this->db->like('ID_CALON',$id,'after');
        $query=$this->db->get();
        if($query->row_array()>0)
        {
            return $query->result();
        }

    }
    public function get_calon_byId($id)
    {
        $this->db->from($this->tab calon);
        $this->db->where('ID_CALON',$id);
        $this->db->limit(1);
        $query = $this->db->get();
        if($query->row_array()>0)
        {
            return $query->result();
        }
    }

    public function get_suara_calon($no,$id)
    {
        $this->db->from($this->tab_suara2);
        $this->db->where('ID_CALON',$no);
```

```
        $this->db->like('ID_TPS',$id,'after');
        $query=$this->db->get();
        if($query->row_array()>0)
        {
            return $query->result();
        }
    }
    public function get_total_suara_calon_masuk($id)
    {
        $this->db->select_sum('TOTAL_SUARA_CALON');
        $this->db->like('ID_TPS',$id,'after');
        $this->db->from($this->tab_suara2);
        $query = $this->db->get();

            return $query->result();
    }
    public function get_jumlah_suara_calon($no,$id)
    {
        $this->db->select_sum('TOTAL_SUARA_CALON');
        $this->db->where('ID_CALON',$no);
        $this->db->like('ID_TPS',$id,'after');
        $this->db->from($this->tab_suara2);
        $query = $this->db->get();

            return $query->result();
    }
    public function get_jumlah_tps_Daerah($id)

    {
        $this->db->like('ID_TPS',$id,'after');
        $this->db->from($this->tab_suara2);
        $query = $this->db->get();

        return $query->num_rows();
    }
    public function get_admin($id,$level)
    {
        $this->db->like('ID_TPS',$id,'after');
        $this->db->where('LEVEL_ADMIN',$level);
        $this->db->from($this->table);
        $query = $this->db->get();
        if($query->num_rows())
        {
            return $query->result() ;
        }

    }
    public function add_admin($data)
    {
        $this->db->insert($this->table, $data);
```

```
        return true;
    }
    public function cek_lock_admin($id)
    {
        $cek ="";
        $this->db->select('locked_status');
        $this->db->where("ID_ADMIN",$id);
        $this->db->limit(1);
        $query = $this->db->get($this->table);
        if($query->row_array()>0)
        {
            foreach ($query->result() as $key) {
                $cek = $key->locked_status;
            }
            return $cek ;
        }
    }
    public function update_data_admin($id,$data)
    {
        $this->db->where('ID_ADMIN',$id);
        $this->db->update($this->table,$data);
        return true ;
    }
    public function add_log_admin($data)
    {
        $this->db->insert($this->tab_log, $data);
        return true;
    }

}
```

3. Kelas Admin Controller (admin.php)

```
<?php if (!defined('BASEPATH')) die();
class Admin_kota extends MY_Controller {
    function __construct()
{
    parent::__construct();
    if(!$userdata = $this->session->userdata('logged_in'))
    {
        redirect('admin');
    }
    $this->load->model('admin_model');

}
    public $tempid="";
    public function cek_admin()
    {

        if ($this->cek_session() == "Kabupaten/Kota")
        {
```

```
            return true;
        }
        else {
            //$this->akses dilarang();
            $this->session->unset_userdata("logged_in");
            $this->session->sess_destroy();
            redirect('admin');

        }
    }
    public function index()
    {
        $this->cek admin();
        $data['admin'] = $this->get namaadmin();
        $data['level']=$this->get leveladmin();
        $id = $this->get_idadmin();
        $this->tempid  = substr($id,2,-3);
        $data['n_kota'] = $this->admin_model-
>get kab kota($this->tempid);
        $this->load->view('include/header',$data);
        $this->load->view('kota/view_adminKota',$data);
        $this->load->view('include/footer');
      }
        public function get_panitia()
    {
        $this->cek admin();
        $data['admin'] = $this->get_namaadmin();
        $data['level']=$this->get_leveladmin();

        $id  = $this->get_idadmin();
        $this->tempid  = substr($id,2,-3);
        $data['panitia'] = $this->admin_model-
>get_panitia($this->tempid);

        $this->load->view('include/header',$data);
        $this->load->view('kota/view kotaPanitia',$data);
        $this->load->view('include/footer');
    }
    public function form_panitia()
    {
        $this->cek admin();
        $data['admin'] = $this->get namaadmin();
        $data['level']=$this->get leveladmin();
        //ambil variable url
        $mau_ke = $this->uri->segment(3);
        $idu    = $this->uri->segment(4);

        $id              = addslashes($this->input-
>post('id'));
```

```
        $nomer          = addslashes($this->input-
>post('nomer'));
        $nama           = addslashes($this->input-
>post('nama'));
        $id_tps         = addslashes($this->input-
>post('id_tps'));
        $status         = addslashes($this->input-
>post('status'));
        if ($mau_ke == "add") {//jika uri segmentnya add
            $data['title'] = 'Tambah panitia';
            $data['aksi'] = 'aksi_add';
            $this->load->view('include/header',$data);
            $this->load->view('kota/view_addPanitia',$data);
            $this->load->view('include/footer');
        } else if ($mau_ke == "edit") {//jika uri segmentnya
edit
            $data['qdata']  = $this->admin_model-
>get_data_panitia($idu);
            $data['title'] = 'Edit';
            $data['aksi'] = 'aksi_edit';
            $this->load->view('include/header',$data);
            $this->load->view('kota/view_addPanitia',$data);
            $this->load->view('include/footer');
        } else if ($mau_ke == "aksi_add") {//jika uri
segmentnya aksi add sebagai fungsi untuk insert
            $aksiadd = "Add Panitia";
            $data = array(
                'ID_PANITIA'        => $id,
                'NMR_URUT_PARTAI'   => $nomer,
                'NAMA_PANITIA'      => $nama,
                'ID_TPS'            => $id_tps,
                'STATUS_PANITIA'    => $status,
            );
            $datalog = array('Id_log' =>$this->get_id_log(),
                             'id_admin'=>$this->get_idadmin(),
                             'aksi'=> $aksiadd,
                             'ip'=> $this->get_ip(),
                             'Id_data'=> $id,
                             'waktu'=> $this->get_date()
                             );
            $this->admin_model->add_log_admin($datalog);
            $this->admin_model->insert_data_panitia($data);
//model insert
            $this->session->set_flashdata("pesan", "<div
class=\"alert alert-success\" id=\"alert\"><i
class=\"glyphicon glyphicon-ok\"></i> Data berhasil di
insert</div>"); //pesan yang tampil setalah berhasil di
insert
            redirect('Admin_kota/get_panitia');
```

```
        } else if ($mau_ke == "aksi_edit") { //jika uri
segmentnya aksi edit sebagai fungsi untuk update
            $aksiedit = "Edit Panitia";
            $data = array(
                    'ID_PANITIA'          => $id,
                    'NMR_URUT_PARTAI'     => $nomer,
                    'NAMA_PANITIA'        => $nama,
                    'ID_TPS'              => $id_tps,
                    'STATUS_PANITIA'              => $status,
                );
            $datalog = array('Id_log' =>$this->get_id_log(),
                             'id_admin'=>$this->get_idadmin(),
                             'aksi'=> $aksiedit,
                             'ip'=> $this->get_ip(),
                             'Id_data'=> $id,
                             'waktu'=> $this->get_date()
                             );
           $this->admin_model->add_log_admin($datalog);
              $this->admin_model-
>update_data_panitia($id,$data); //modal update data barang
              $this->session->set_flashdata("pesan", "<div
class=\"alert alert-success\" id=\"alert\"><i
class=\"glyphicon glyphicon-ok\"></i> Data berhasil
diupdate</div>"); //pesan yang tampil setelah berhasil di
update
              redirect('Admin_kota/get_panitia');
        }
    }
    public function detail_panitia($id){ //fungsi detail
Panitia
        $this->cek_admin();
        $data['admin'] = $this->get_namaadmin();
        $data['level']=$this->get_leveladmin();
        $data['title'] = 'Detail Panitia';
        $data['qdata'] = $this->admin_model-
>get_data_panitia($id); //query model barang sesuai id
        $this->load->view('include/header',$data);
        $this->load->view('kota/view_detPanitia',$data);
//meload views detail barang
        $this->load->view('include/footer');
    }
    public function hapus($gid){
        $aksi = "Hapus Panitia";
        $this->admin_model->delete_data_panitia($gid);
        $datalog = array('Id_log' =>$this->get_id_log(),
                         'id_admin'=>$this->get_idadmin(),
                         'aksi'=> $aksi,
                         'ip'=> $this->get_ip(),
                         'Id_data'=> $gid,
                         'waktu'=> $this->get_date()
```

```
                                );
            $this->admin model->add log admin($datalog);
        $this->session->set flashdata("pesan", "<div
class=\"alert alert-danger\" id=\"alert\"><i
class=\"glyphicon glyphicon-ok\"></i> Data berhasil
dihapus</div>");
        redirect('Admin_kota/get_panitia');

    }

    public function validasi_pemilih()
    {
        $this->cek admin();
        $data['admin'] = $this->get namaadmin();
        $data['level']=$this->get leveladmin();

        $id  = $this->get_idadmin();
        $this->tempid  = substr($id,2,-3);

        $data['pemilih'] = $this->admin model-
>get_pemilih($this->tempid);
        $this->load->view('include/header',$data);
        $this->load->view('kota/view_validasi',$data);
        $this->load->view('include/footer');
    }

    public function set_validasi($id)
    {
        $statusOk ="1";
        $statusNo ="0";
        $aksi = "Validasi Pemilih";

        $hasil = $this->admin_model-
>cek validasi pemilih($id);
        if($hasil == 1){
        $data = array(
                'Status'                => $statusNo
            );
        }
        elseif($hasil == 0){
        $data = array(
                'Status'                => $statusOk
            );
        }

        $datalog = array('Id_log' =>$this->get_id_log(),
                         'id admin'=>$this->get idadmin(),
                         'aksi'=> $aksi,
                         'ip'=> $this->get_ip(),
                         'Id_data'=> $id,
```

```
                            'waktu'=> $this->get date()
                            );
        $this->admin model->add log admin($datalog);

        $this->admin_model->update_data_pemilih($id,$data);
        $this->session->set_flashdata("pesan", "<div
class=\"alert alert-success\" id=\"alert\"><i
class=\"glyphicon glyphicon-ok\"></i> Status Validasi
Berhasil Di Ubah</div>"); //pesan yang tampil setelah
berhasil di update
        redirect('Admin_kota/validasi_pemilih');
    }

    public function set_all_validasi()
    {
        $statusOk ="1";
        $aksi = "set Beberapa Pemilih";
        $nik = array();
        //$hasil = $this->admin model-
>cek validasi pemilih(implode("','",array values($this-
>input->post['checkbox'])));
        $nik = $this->input->post('checkbox');
        $data = array(
                'Status'                => $statusOk
            );
        $datalog = array('Id_log' =>$this->get_id_log(),
                            'id_admin'=>$this->get_idadmin(),
                            'aksi'=> $aksi,
                            'ip'=> $this->get_ip(),
                            'Id_data'=> $nik,
                            'waktu'=> $this->get date()
                            );
        $this->admin_model->add_log_admin($datalog);
        $this->admin model-
>update all data pemilih($nik,$data);
        $this->session->set flashdata("pesan", "<div
class=\"alert alert-success\" id=\"alert\"><i
class=\"glyphicon glyphicon-ok\"></i> Status Validasi Data
Yang Dipilih Berhasil Di Ubah</div>"); //pesan yang tampil
setelah berhasil di update
        redirect('Admin_kota/validasi_pemilih');
    }
    public function cek_validasi($nik)
    {

        $hasil = $this->admin model-
>cek validasi pemilih($nik);
        if($hasil == 1 )
        {
```

```
            $this->session->set_flashdata("pesan", "<div
class=\"alert alert-success\" id=\"alert\"><i
class=\"glyphicon glyphicon-ok\"></i> Data Sudah
Tervalidasi</div><a href=\"a\"></a>");
            redirect('Admin_kota/validasi_pemilih');
        }
        else
        {
            $this->session->set_flashdata("pesan", "<div
class=\"alert alert-danger\" id=\"alert\"><i
class=\"glyphicon glyphicon-ok\"></i> Data Belum
Divalidasi</div>");
            redirect('Admin_kota/validasi_pemilih');
        }

    }
    public function get_pemilih()
    {

        $this->cek_admin();
        $data['admin'] = $this->get_namaadmin();
        $data['level']=$this->get_leveladmin();

        $id  = $this->get_idadmin();
        $this->tempid  = substr($id,2,-3);

        $data['pemilih'] = $this->admin_model-
>get_pemilih($this->tempid);
        $this->load->view('include/header',$data);
        $this->load->view('kota/view_kotaPemilih',$data);
        $this->load->view('include/footer');
    }
    public function form_pemilih()
    {
        $this->cek_admin();
        $data['admin'] = $this->get_namaadmin();
        $data['level']=$this->get_leveladmin();
        //ambil variable url
        $mau_ke = $this->uri->segment(3);
        $idu    = $this->uri->segment(4);
        $this->load->library('upload');
        $nmfile = "file ".time(); //nama file saya beri nama
langsung dan diikuti fungsi time
        $config['upload_path'] = './assets/uploads/'; //path
folder
        $config['allowed_types'] = 'gif|jpg|png|jpeg|bmp';
//type yang dapat diakses bisa anda sesuaikan
        $config['max_size'] = '2048'; //maksimum besar file
2M
```

```
        $config['max width']  = '1288'; //lebar maksimum
1288 px
        $config['max height']  = '768'; //tinggi maksimu 768
px
        $config['file_name'] = $nmfile; //nama yang
terupload nantinya
        $this->upload->initialize($config);

        $nik                = addslashes($this->input-
>post('nik'));
        $id_tps             = addslashes($this->input-
>post('id_tps'));
        $nama               = addslashes($this->input-
>post('nama'));
        $tempat_lahir       = addslashes($this->input-
>post('tempat_lahir'));
        $tanggal_lahir      = addslashes($this->input-
>post('tanggal_lahir'));
        $jenis_kelamin      = addslashes($this->input-
>post('jenis_kelamin'));
        $alamat             = addslashes($this->input-
>post('alamat'));
        //$status             = addslashes($this->input-
>post('status'));
        $agama              = addslashes($this->input-
>post('agama'));
        //$foto               = addslashes($this->upload-
>do_upload('foto'));
        if ($mau_ke == "add") {//jika uri segmentnya add
            $data['title'] = 'Tambah panitia';
            $data['aksi'] = 'aksi_add';
            $this->load->view('include/header',$data);
            $this->load->view('kota/view_addPemilih',$data);
            $this->load->view('include/footer');
        } else if ($mau_ke == "edit") {//jika uri segmentnya
edit
            $data['qdata']  = $this->admin_model-
>get_data_pemilih($idu);
            $data['title'] = 'Edit';
            $data['aksi'] = 'aksi_edit';
            $this->load->view('include/header',$data);
            $this->load->view('kota/view_addPemilih',$data);
            $this->load->view('include/footer');
        } else if ($mau_ke == "aksi_add") {//jika uri
segmentnya aksi_add sebagai fungsi untuk insert
            $aksiadd = "Add Pemilih";
            $data = array(
                'NIK'               => $nik,
                'ID_TPS'            => $id_tps,
                'NAMA_LENGKAP'      => $nama,
```

```
                'TEMPAT LAHIR'        => $tempat_lahir,
                'TANGGAL LAHIR'       => $tanggal_lahir,
                'JENIS KELAMIN'       => $jenis_kelamin,
                'ALAMAT LENGKAP'      => $alamat,
                'AGAMA'               => $agama,
                'FOTO'                => $this->upload-
>do_upload(),
            );
            $datalog = array('Id_log' =>$this->get_id_log(),
                        'id_admin'=>$this->get_idadmin(),
                        'aksi'=> $aksiadd,
                        'ip'=> $this->get_ip(),
                        'Id_data'=> $nik,
                        'waktu'=> $this->get_date()
                        );
            $this->admin_model->add_log_admin($datalog);
            $this->admin_model->insert_data_pemilih($data);
//model insert
            $this->session->set_flashdata("pesan", "<div
class=\"alert alert-success\" id=\"alert\"><i
class=\"glyphicon glyphicon-ok\"></i> Data berhasil di
insert</div>"); //pesan yang tampil setalah berhasil di
insert
            redirect('Admin_kota/get_pemilih');
        } else if ($mau_ke == "aksi_edit") { //jika uri
segmentnya aksi edit sebagai fungsi untuk update
            $aksiedit = "Edit Pemilih";
            $data = array(
                'NIK'                 => $nik,
                'ID_TPS'              => $id_tps,
                'NAMA LENGKAP'        => $nama,
                'TEMPAT_LAHIR'        => $tempat_lahir,
                'TANGGAL_LAHIR'       => $tanggal_lahir,
                'JENIS KELAMIN'       => $jenis_kelamin,
                'ALAMAT LENGKAP'      => $alamat,
                'AGAMA'               => $agama,
                'FOTO'                => $this->upload-
>do_upload(),
            );
            $datalog = array('Id_log' =>$this->get_id_log(),
                        'id_admin'=>$this->get_idadmin(),
                        'aksi'=> $aksiedit,
                        'ip'=> $this->get_ip(),
                        'Id_data'=> $nik,
                        'waktu'=> $this->get_date()
                        );
            $this->admin_model->add_log_admin($datalog);
            $this->admin_model-
>update_data_pemilih($id,$data);
```

```
            $this->session->set flashdata("pesan", "<div
class=\"alert alert-success\" id=\"alert\"><i
class=\"glyphicon glyphicon-ok\"></i> Data berhasil
diupdate</div>"); //pesan yang tampil setelah berhasil di
update
            redirect('Admin_kota/get_pemilih');
        }
    }
    public function detail_pemilih($id){
        $this->cek_admin();
        $data['admin'] = $this->get_namaadmin();
        $data['level']=$this->get_leveladmin();
        $data['title'] = 'Detail Panitia';
        $data['qdata'] = $this->admin model-
>get data pemilih($id);
        $this->load->view('include/header',$data);
        $this->load->view('kota/view_detPemilih',$data);
        $this->load->view('include/footer');
    }
    public function rekapPartai()
    {
        $this->cek_admin();
        $data['admin'] = $this->get_namaadmin();
        $data['level']=$this->get_leveladmin();
        $data['partai'] = $this->admin model->get partai();
        $id  = $this->get idadmin();
        $this->tempid  = substr($id,2,-3);
        $data['hasil'] = $this->admin_model-
>get_suara($this->tempid);
        $this->load->view('include/header',$data);
        $this->load->view('kota/view rekapPartai',$data);
        $this->load->view('include/footer');
    }
    public function partai()
    {
        $this->cek admin();
        $data['admin'] = $this->get namaadmin();
        $data['level']=$this->get_leveladmin();
        $no    = $this->uri->segment(3);
        $hasil = $this->admin model->get partai byNo($no);
        foreach ($hasil as $key ) {
            $data['nm partai'] = $key->NAMA PARTAI;
            $data['lam partai'] = $key->LAMBANG PARTAI;
        }
        //calon partai
        $id  = $this->get idadmin();
        $this->tempid  = substr($id,2,-3);
        $data['n kota'] = $this->admin model-
>get_kab_kota($this->tempid);
        $id_pem = explode(".", $this->tempid);
```

```
        $x = $id_pem[0].$id_pem[1];
        $data['suara_masuk']=$this->admin_model-
>get_total_suara_masuk($this->tempid);
        $data['total_tps']=$this->admin_model-
>get_jumlah_tps_tiapDaerah($this->tempid);
        $data['calon']=$this->admin_model-
>get_calon_partai($no,$x);
        $data["total_suara"] = $this->admin_model-
>get_jumlah_suara_partai($no,$this->tempid);
        $data['suara'] = $this->admin_model-
>get_suara_partai($no,$this->tempid);
        $this->load->view('include/header',$data);
        $this->load->view('kota/view_partai',$data);
        $this->load->view('include/footer');
    }
    public function detail_calon()
    {
        $this->cek_admin();
        $idx  = $this->get_idadmin();
        $this->tempid  = substr($idx,2,-3);
        $data['n_kota'] = $this->admin_model-
>get_kab_kota($this->tempid);
        $id = $this->uri->segment(3);
        $data['admin'] = $this->get_namaadmin();
        $data['level']=$this->get_leveladmin();
        $data['title'] = 'Detail Calon';
        $hasil = $this->admin_model->get_calon_byId($id);
          foreach($hasil as $rowdata){
        $data['id']            = $rowdata->ID_CALON;
        $data['no_partai']     = $rowdata->NMR_URUT_PARTAI;
        $data['no_calon']      = $rowdata->NMR_URUT_CALON;
        $data['nama']          = $rowdata->NAMA_CALON;
        $data['slogan']        = $rowdata->SLOGAN_CALON;
        $data['foto']          = $rowdata->FOTO_CALON;
        $data['ket']    = $rowdata->KETERANGAN_CALON;
        }
        $result = $this->admin_model-
>get_partai_byNo($data['no_partai']);
        foreach ($result as $key ) {
            $data["nm_partai"] = $key->NAMA_PARTAI;
            $data['lam_partai'] = $key->LAMBANG_PARTAI;
        }
        $data['suara'] = $this->admin_model-
>get_suara_calon($id,$this->tempid);
        $data['suara_masuk']=$this->admin_model-
>get_total_suara_calon_masuk($this->tempid);
        $data['total_tps']=$this->admin_model-
>get_jumlah_tps_Daerah($this->tempid);
        $data["total_suara"] = $this->admin_model-
>get_jumlah_suara_calon($id,$this->tempid);
```

```
        $this->load->view('include/header',$data);
        $this->load->view('kota/view_detCalon',$data);
        $this->load->view('include/footer');
    }
    public function form_admin()
    {
        $this->cek_admin();
        $data['admin'] = $this->get_namaadmin();
        $data['level']=$this->get_leveladmin();
        $idx  = $this->get_idadmin();
        $this->tempid  = substr($idx,2,-3);
        $data['kecamatan'] = $this->admin_model-
>get_kecamatan_kodeKota($this->tempid);
        $data['tps'] = $this->admin_model-
>get_TPS_kodeDesa($this->tempid);
        $this->load->view('include/header',$data);
        $this->load->view('kota/view_addAdmin',$data);
        $this->load->view('include/footer');
    }
    public function get_admin()
    {
        $this->cek_admin();
        $data['admin'] = $this->get_namaadmin();
        $data['level']=$this->get_leveladmin();

        $id  = $this->get_idadmin();
        $this->tempid  = substr($id,2,-3);
        $level1 = "Kecamatan";
        $data['dataadmin'] = $this->admin_model-
>get_admin($this->tempid,$level1);
        $this->load->view('include/header',$data);
        $this->load->view('kota/view_kotaAdmin',$data);
        $this->load->view('include/footer');
    }
    public function buka_lock($id)
    {
       $lockyes = "yes";
       $lockno = "no";
       $aksi = "Ubah Lock Status Admin";
        $hasil = $this->admin_model->cek_lock_admin($id);
        if($hasil == yes){
        $data = array(
                'locked_status'                 => $lockno
            );
        }
        elseif($hasil == 0){
        $data = array(
                'locked_status'                 => $lockyes
            );
```

```
        }

        $datalog = array('Id_log' =>$this->get_id_log(),
                         'id admin'=>$this->get idadmin(),
                         'aksi'=> $aksi,
                         'ip'=> $this->get_ip(),
                         'Id_data'=> $id,
                         'waktu'=> $this->get date()
                         );
        $this->admin_model->add_log_admin($datalog);

        $this->admin_model->update_data_admin($id,$data);
        $this->session->set flashdata("pesan", "<div
class=\"alert alert-success\" id=\"alert\"><i
class=\"glyphicon glyphicon-ok\"></i> Lock Status Berhasil
Di Ubah</div>"); //pesan yang tampil setelah berhasil di
update
        redirect('Admin_kota/get_admin');
    }
    public function add_admin()
    {
        $this->cek_admin();
        $data['admin'] = $this->get_namaadmin();
        $data['level']=$this->get_leveladmin();
        $idx  = $this->get idadmin();
        $this->tempid  = substr($idx,2,-3);
        $id            = addslashes($this->input-
>post('id'));
        $id_tps        = addslashes($this->input-
>post('id_tps'));
        $nama          = addslashes($this->input-
>post('nama'));
        $password      = addslashes($this->input-
>post('password'));
        $level         = addslashes($this->input-
>post('level'));
        $token              =rand();
        $hasil = $this->enkrip_password($password,$id);
        //$keterangan   = addslashes($this->input-
>post('keterangan'));
        $aksi = "Add Admin";
        $data = array(
                'ID_ADMIN'              => $id,
                'ID_TPS'            => $id_tps,
                'NAMA_ADMIN'      => $nama,
                'PASSWORD ADMIN'      => ($hasil),
                'LEVEL_ADMIN'     => $level,
                'token'             => $token,
            );
        $datalog = array('Id_log' =>$this->get_id_log(),
```

```
                                'id admin'=>$this->get idadmin(),
                                'aksi'=> $aksi,
                                'ip'=> $this->get ip(),
                                'Id data'=> $id,
                                'waktu'=> $this->get_date()
                                );
        $this->admin_model->add_log_admin($datalog);
        $this->admin model->add admin($data);
        $this->session->set flashdata("pesan", "<div
class=\"alert alert-success\" id=\"alert\"><i
class=\"glyphicon glyphicon-ok\"></i> Data berhasil di
Tambah</div>");
        redirect('Admin kota/get admin');
    }
    public function backup_database()
    {
        $this->cek_admin();
        $this->load->dbutil();
        $nama = $this->get namaadmin();
        $prefs = array(
                'tables'        => array('panitia',
'partai','pemilih','temp_suara_partai','temp_suara_partai'),
                'format'        => 'zip',
                'filename'      => 'my_db_backup.sql'
                );


        $backup =& $this->dbutil->backup($prefs);

        $db_name = $nama.'backup-on-'. date("Y-m-d-H-i-s")
.'.zip';
        $save = 'pathtobkfolder/'.$db_name;

        $this->load->helper('file');
        write file($save, $backup);


        $this->load->helper('download');
        force_download($db_name, $backup);
    }

    public function statPartai()
    {
        $this->cek_admin();
         $id  = $this->get_idadmin();
        $data['admin'] = $this->get namaadmin();
        $data['level']=$this->get leveladmin();
        $this->tempid  = substr($id,2,-3);
        $data['n_kota'] = $this->admin_model-
>get_kab_kota($this->tempid);
```

```
        $data['hasil'] = $this->admin model-
>get suara($this->tempid);

        $this->load->view('include/header',$data);
        $this->load->view('kota/view_statPartaiKota',$data);
        $this->load->view('include/footer');
    }
    public function statCalon()
    {

        $this->cek_admin();
         $id  = $this->get_idadmin();
        $data['admin'] = $this->get namaadmin();
        $data['level']=$this->get leveladmin();
        $this->tempid  = substr($id,2,-3);
        $data['n_kota'] = $this->admin_model-
>get_kab_kota($this->tempid);
        $data['hasil'] = $this->admin_model-
>get suaraCalon($this->tempid);
        $this->load->view('include/header',$data);
        $this->load->view('kota/view_statCalonKota',$data);
        $this->load->view('include/footer');
    }


}
```

DAFTAR PUSTAKA

[1] Anonim, “Electronic Voting atau E-Voting,” Rumah Pemilu, [Online]. Available: http://www.rumahpemilu.org/in/read/15/Electronic-Voting-atau-E-Voting. [Diakses 28 Desember 2014].

[2] K. D. N. R. Indonesia, “Desa Mendoyo, Contoh "e-Voting" Pertama,” 31 Juli 2013. [Online]. Available: http://otda.kemendagri.go.id/index.php/categoryblog/1072-desa-mendoyo-contoh-qe-votingq-pertama. [Diakses 21 Juni 2015].

[3] A. Rokhman, “Prospek Penerapan E-Voting di Indonesia,” Magister Administrasi Publik Universitas Jenderal Soedirman Purwokerto, 29 November 2011. [Online]. Available: http://map.unsoed.ac.id/2011/11/29/prospek-penerapan-e-voting-di-indonesia/. [Diakses 29 Desember 2014].

[4] R. Sadikin, Kriptografi untuk Keamanan Jaringan, Yogyakarta: Andi, 2012.

[5] php, “php,” [Online]. Available: http://php.net/manual/en/intro-whatis.php. [Diakses 20 12 2015].

[6] OWASP, “OWASP,” 01 8 2015. [Online]. Available: https://www.owasp.org/index.php/Cross-Site_Request_Forgery_%28CSRF%29_Prevention_Cheat_Sheet. [Diakses 1 10 2015].

[7] EllisLab, “Codeigniter,” [Online]. Available: https://ellislab.com/codeigniter/user-guide/. [Diakses 07 10 2015].

[8] Highcharts, “Highcharts,” [Online]. Available: http://www.highcharts.com/products/highcharts. [Diakses 12

October 2015].

[9] sprymedia, “datatables,” 01 08 2007. [Online]. Available: http://datatables.net/. [Diakses 04 11 2015].

[10] K. Puguh, Implementasi Kontrol Integritas Kertas Suara Untuk Pengamanan Sistem pemungutan Suara Secara Elektronik (E-VOTING), Surabaya, 2015.

BIODATA PENULIS

Ahmad Natsir Fauzi, yang biasa dipanggil Fauzi atau Ujek, penulis dari buku Tugas Akhir ini, lahir di Sidoarjo pada tanggal 19 April 1993. Penulis adalah anak ketiga dari lima bersaudara. Penulis telah menempuh pendidikan di TK TPI (Taman Pendidikan Islam) Gedangan, Sidoarjo (1997-1999), SDN Ketajen II Gedangan, Sidoarjo (1999-2005), SMP Negeri 3 Sidoarjo (2005-2008), SMA Negeri 1 Sidoarjo (2008-2011), dan terakhir di S1 Teknik Informatika ITS Surabaya (2011-2015).

Selama masa perkuliahan penulis pernah aktif sebagai anggota organisasi Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) ITS. Penulis memilih bidang minat Arsitektur Jaringan Komputer (AJK) dan tertarik web develpoer. Penulis memiliki hobi membaca dan futsal.

Penulis dapat dihubungi melalui alamat email: ujekfauzy@gmail.com